Unaligned Message-Passing and Contextualized-Pretraining for Robust Geo-Entity Resolution

Yuwen Ji[1,2], Wenbo Xie[1*], Jiaqi Zhang[1], Chao Wang[1], Ning Guo[1], Lei Shi[2], Yue Zhang[3]
[1]Amap, China
[2]Beihang University, China
[3]Westlake University, China
daoqian.xwb@alibaba-inc.com, {jiyuwen, leishi}@buaa.edu.cn, zhangyue@westlake.edu.cn

Abstract

Geo-entity resolution involves linking records that refer to the same entities across different spatial datasets, which underpins location-based services. Given the varying quality of geo-data, this task is known to be challenging, as directly comparing the semantic-centric representations of two entities is no longer reliable. To robustify geo-entity resolution in this context, the main research question is how to effectively extend the current semantics-centric representations of geo-entity with geographical context from its spatial neighbors. Existing methods consider names from neighbors, but they struggle to fully utilize the unaligned neighbor attributes. In this paper, we study the representation of geo-context for robust geo-entity resolution and propose two adaptations that efficiently leverage unaligned geo-entity attributes across spatial neighbors: (1) A plugin module, namely Unaligned Message-Passing (UMP), that propagates unaligned neighbor features to integrate geo-context into the token embeddings output by language model; (2) a contextualized pretraining framework (CP) that allows the former to leverage unlabelled geo-entity data. Experiments show that our method surpasses the baselines, achieving higher F1 scores on 8 real-world geo-datasets in terms of robustness, with an improvement of up to 7.9%. The ablation study further justifies our proposal.

Introduction

Geo-entity resolution (GER) (Sehgal, Getoor, and Viechnicki 2006) involves linking records that refer to the same entities across spatial datasets, which underpins location-based services. As shown in Figure 1, a geo-entity record is typically identified by textual attributes (name, address, etc.) and the spatial position (latitude & longitude). The task is to determine whether the geo-entity records from one source (red table) match the records from another source (green table). Typically, GER consists of two steps: (1) extracting underlying features to create a robust record representation for generic understanding, and (2) classifying whether two records represent same entity based on their representations.

This work focuses on the first step, as a better understanding of geo-entities aids in matching. The varying quality of geospatial data, such as aliases, spelling errors, and missing attributes, complicates direct semantic comparison, especially for nearby entities. Previous work (Balsebre et al. 2022) has shown that entities with similar names and locations might not match due to inaccuracies of geo-positional systems or being chain brands like Starbucks in a mall. However, incorporating neighboring entities can assist us in making such inferences. For instance, the same-name pair in Figure 1(a) is not a match due to different neighbors (blue icon). Pre-trained transformer-based language models (Vaswani et al. 2017) effectively represent geo-entities by capturing deep semantic relationships, outperforming traditional rule-based methods. The research question is how to extend these semantics-centric representations with geo-context from spatial neighbors to enhance GER robustness.

In the field of GER, methods have evolved from earlier rule-based approaches (Deng et al. 2019; Shivaprabhu, Balasubramani, and Cruz 2017; Miller et al. 1990; Morana et al. 2014; Isaj, Zimányi, and Pedersen 2019) to emerging deep learning-based solutions (Li et al. 2023, 2022; Tempelmeier, Gottschalk, and Demidova 2021; Balsebre et al. 2022, 2023a). These methods primarily focus on comparing semantic aspects between entities, often neglecting geo-context, w.r.t. spatial neighbors. As a result, they rely heavily on clean and complete records to construct accurate attribute representations, and their performance suffers with even minor data quality issues, like spelling errors or missing values. We argue that leveraging geo-context can mitigate performance degradation in noisy datasets. For instance, erroneous or missing values can be corrected by examining fine-grained tokens of neighbors (Figure 1(b)), and having common neighbors (e.g., *Beer* in blue icon) can serve as a matching criterion. Although some works, such as SpaBERT (Li et al. 2022) and Geo-ER (Balsebre et al. 2022), have considered neighbor names, they fail to fully utilize the fine-grained tokens of neighbor attributes. To the best of our knowledge, our problem remains underexplored.

Towards robust geo-entity resolution, we use pre-trained language models to construct attribute representations and graph neural networks (GNNs) to incorporate geographical context from neighboring entities. The motivation is that the unordered message-passing mechanism of GNNs (Kipf and Welling 2016; Dwivedi et al. 2020; Wu et al. 2020) is well-suited for this task as it aligns with the non-sequential propagation of attributes from neighbors. Despite advancements in semantics-centric representations with language

*Corresponding author

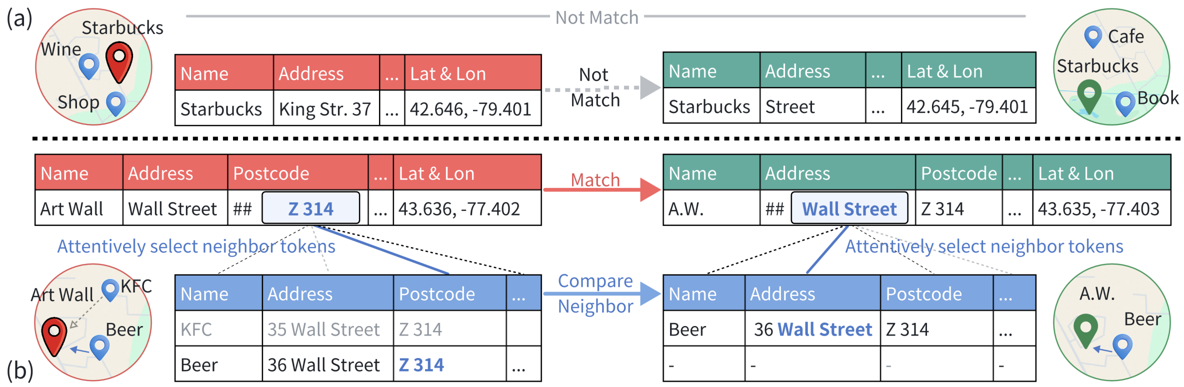

Figure 1: The challenges in geo-entity resolution. (a) The same-name candidate pair is not a match because they have different neighbors (in blue icon); (b) we correct erroneous/missing values (denoted by ##) by taking the fine-grained tokens of neighbors (in blue table), having the same neighbors (*Beer* in blue icon) also serves as a matching criterion.

models, challenges remain in robust GER. First, existing GNNs struggle with the unaligned 2-dimensional features encoded by language models; yet the GER requires selective token comparison without naive pooling to avoid losing fine-grained information (Figure 1). Second, the potential of unlabeled geo-entity data is untapped due to the lack of pre-training tasks designed for capturing geographical context. Lastly, evaluating robustness is a new task that necessitates examining the utility of existing datasets and metrics.

Overall, our contributions are summarized as follows:

- We adopt a BERT-based encoder for geo-entity representation. To efficiently inject geo-context, a module of Unaligned Message-Passing (UMP) is introduced.
- We propose a contextualized pre-training (CP) with 2 tasks, allowing the former module to leverage unlabeled data and capture geographical context. Moreover, we draw a connection to an information theory.
- We develop a method for evaluating the robustness of various GER methods using diverse real-world datasets.

Experimentally, our F1 scores are constantly ranked in the top-1 over 8 real-world GER datasets, with an improvement up to 7.9%, showcasing the superiority of our proposal. The ablation study also confirms the usefulness of our design.

Related Works

Geo-Entity Resolution

The first type of method based on pre-training utilizes large-scale, unlabeled data to train general representations and fine-tune them for GER. Notable examples include Geo-ER (Balsebre et al. 2022), which uses a pre-trained language model to encode entity attributes, and a graph neural network (GNN) to integrate neighbor names. However, Geo-ER's GNN application is incompatible with the language model's word embeddings, requiring pooling before passing neighbor messages, which can lose fine-grained information (Stergiou and Poppe 2021; Lang et al. 2023). After pooling, it is hard to denoise attribute tokens, leading to performance degradation with noisy data. Following Geo-ER, GTMiner (Balsebre et al. 2023b) is proposed for better knowledge graph construction performance. SpaBERT (Li et al. 2022) concatenates neighbor names into a serialized sentence, adding longitude and latitude to the position encoding to capture geo-context. It introduces a masked language modeling pre-training task to utilize neighbor names for recovering masked tokens. However, SpaBERT focuses on names and encodes neighbor information in a serialized manner, despite the unordered nature of attribute propagation from neighboring entities. GeoVectors (Tempelmeier, Gottschalk, and Demidova 2021) uses 2 embedding models to encode positional and semantic information, inferring representations of unseen entities by combining existing neighbors' embeddings through a weighted average. Yet, it does not inject geo-context into the representations.

On the other hand, a few pretraining-based methods involve a bridging process to align crude attribute representations with mature unimodal representations. GeoLM (Li et al. 2023) uses contrastive learning on top of SpaBERT to align (geo-entity, article) pairs from Wikidata and Wikipedia but shares SpaBERT's limitations. CITYFM (Balsebre et al. 2023a) aligns multimodal data (entity and geometry image). Earlier methods, such as location-based contributions (Abdalla 2016; Balley, Parent, and Spaccapietra 2004; Tabarro et al. 2017; Schäfers and Lipeck 2014), primarily relied on coordinates to determine entity matches. Rule-based solutions (Deng et al. 2019; Shivaprabhu, Balasubramani, and Cruz 2017; Miller et al. 1990; Morana et al. 2014; Isaj, Zimányi, and Pedersen 2019) depended on manually designed rules or textual similarity metrics. Neither of these methods considers attribute semantics and geo-context.

General Entity Resolution

While geo-specific methods that consider spatial correlation have shown promise, general entity resolution (ER) approaches (Christophides et al. 2019; Barlaug and Gulla 2020) can also be applied to geo-entity matching. Declarative rule-based (Dalvi et al. 2013; Elmagarmid et al. 2014; Wang et al. 2011) and crowd-sourcing-based (Gokhale et al. 2014; Wang et al. 2012; Firmani, Saha, and Srivastava 2016) methods offer human oversight but can be less scalable, and require labor, poor on non-structured data (Mudgal et al. 2018). ML-based methods, which train classifiers like SVM (Bilenko and Mooney 2003), active learning (Sarawagi and Bhamidipaty 2002), and MLP (Ebraheem et al. 2017b; Konda et al. 2018), offer scalability and can learn from data without explicit rules, gaining increasing attention.

In ML-based methods, deep learning has emerged as the state-of-the-art for ER due to its ability to dynamically learn distributed representations of entities. Early adopters like DeepER (Ebraheem et al. 2017a) and DeepMatcher (Mudgal et al. 2018) use LSTMs and sequence-aware RNNs with attention, respectively. Subsequent works, such as M2M (Fu et al. 2019), Seq2Seq (Nie et al. 2019), and HierMatcher (Fu et al. 2020), build on DeepMatcher to enhance performance on specific datasets. Pre-trained language models (LMs) like BERT (Devlin et al. 2019), trained on massive text corpora, have achieved higher accuracy in ER. Ditto (Li et al. 2020) treats ER as sequence-pair classification using fine-tuned LMs, introducing domain knowledge highlighting and data augmentation, and supports many pre-trained LMs, making it the best-performing algorithm for general ER benchmarks and suitable for empirical comparison with our method. Subsequent work (Li et al. 2021) applies BERT in siamese networks for efficient blocking and comparison, and JointBERT (Peeters and Bizer 2021) extends this for multi-class matching. However, the aforementioned solutions do not account for the spatial correlation and geographical context.

Preliminary

Geo-Entity Resolution

Let D be a geo-entity dataset with $|D|$ entities, m textual attributes $A = \{A[1], A[2], \dots, A[m]\}$, and two geospatial attributes $\{latitude, longitude\}$. Each geo-entity $e \in D$ consists of attribute values $V = \{e.A[1], e.A[2], \dots, e.A[m], e.latitude, e.longitude\}$. Here, $e.A[k]$ is the k-th attribute value of entity e. Geo-Entity Resolution (GER) aims to identify matched entity pairs that refer to the same real-world entity across all available geospatial datasets. In its simplest form, GER applies a function M that maps each pair of entities descriptions $(e_1, e_2) \in D_1 \times D_2$ to $\{true, false\}$, where $M(e_1, e_2) = true$ means e_1 and e_2 refer to the same real-world entity, and $M(e_1, e_2) = false$ otherwise.

Language Models for Entity Representation

Pre-trained language models (LMs) based on the Transformer (Vaswani et al. 2017), such as BERT (Devlin et al. 2019) or GPT (Brown et al. 2020; OpenAI 2023), exhibit strong semantic expressiveness. The self-attention mechanism allows these models to generate word embeddings, deeply capturing meanings in various contexts. Leveraging this, the geo-entity community increasingly adopts LMs for entity representation by transforming entity attributes into pseudo-sentences. For example, (Li et al. 2020; Ge et al. 2023) construct pseudo-sentences $S(e)$ for each entity e as:

$$S(e) = [\text{COL}]\ A[1]\ \dots\ [\text{COL}]\ A[m]\ [\text{VAL}]\ e.A[m] \quad (1)$$

here the token [COL] precedes the attribute name, and the token [VAL] precedes its value, indicating to the LM model which attribute belongs to a set of tokens. After inputting the sequence $S(e)$ of length $|S(e)|$ into LMs, the output is a sequence of the same token length as the input:

$$h = LM(S(e)) \quad (2)$$

each token is a d-dimensional embedding of the input word, denoted by $h \in \mathbb{R}^{|S(e)| \times d}$. This enables encoding each entity attribute with different lengths and types simultaneously, injecting context information across the entity's attributes.

Message-Passing for Geo-Contextualization

Most current Graph Neural Networks (GNNs) can be formulated as message-passing architecture, where node representations are computed by iteratively aggregating the embeddings of their neighboring nodes. Typically, GNNs embed node n_i through the following equation:

$$o_i^l = \text{AGG}^l(\{(h_j^{l-1}, r_{ij}) : j \in \mathcal{N}(i)\}) \quad (3)$$

$$h_i^l = \text{UPDATE}^l(h_i^{l-1}, o_i^l) \quad (4)$$

h_i^{l-1} and h_i^l are the embeddings of node n_i before and after the l-th convolution, r_{ij} is the embedding of the edge between n_i and n_j, and $\mathcal{N}(i)$ represents the neighbors of n_i. Specifically, Eq. (3) computes the aggregated message o_i^l from $\mathcal{N}(i)$, while Eq. (4) combines o_i^l with h_i^{l-1} to produce the updated embedding h_i^l for node n_i.

In real-world, an entity's latitude and longitude can vary based on user-uploaded POI (point of interest) positions, such as entering campus from different gates. This variability can alter the sequential distance ranking of neighbors relative to the pivot entity. Moreover, spatial distance differs from token distance in the sentence. Thus, neighbor information should not be learned in a serialized manner, it's highly recommend to capture the non-sequential features by passing messages between related entities (i.e. nodes).

Robust GER Through UMP and CP

We first introduce our information theory, formulating the goals of robust GER. Subsequently, we present our framework based on the theory and justify our design step by step.

Information Theory Approach to Robust GER

We denote the real geo-entity with a random variable x_i and the observed part (record) in the geo-dataset with the random variable e_i, where i represents a specific entity. $I(x_i; e_i)$ indicates the mutual information between x_i and e_i, measuring how much information one variable provides about the other. Since x_i is often unavailable, prevailing approaches find matched entity pairs based on pairwise entity comparison, i.e., $I(e_1; e_2)$. Yet, noisy data makes measuring $I(e_1; e_2)$ unreliable. To tackle this, we use e_i and the neighbor entity set $\mathcal{N}(e_i)$ to construct representations z_i that:

$$\max_{z_i} I(x_i; z_i) \quad s.t. \quad I(z_i) - I(x_i; z_i) = 0 \quad (5)$$

where maximizing $I(x_i; z_i)$ extracts entity-related information, zeroing out $(I(z_i) - I(x_i; z_i))$ eliminates noise. We replace $M(e_1, e_2)$ with $M(z_1, z_2)$ for matching in GER.

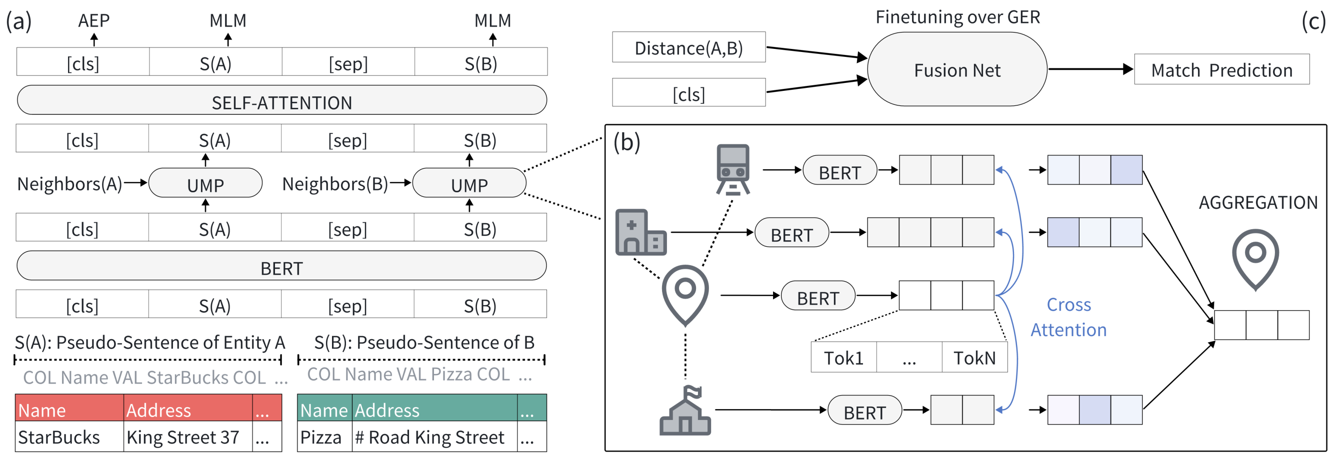

Figure 2: (a) Running example of encoder: The UMP refines the BERT's embeddings, and the self-attention enables communication between the refined tokens, followed by our pretraining tasks (AEP & MLM). (b) UMP overview: Each neighbor's token embeddings serve as keys/values to perform cross-attention with the pivot entity, obtaining aligned messages for aggregation. (c) Finetuning example: pre-trained embedding is used to classify each candidate pair as matching or non-matching.

Framework Overview

As Figure 2 shows, we present our framework that purses robust GER by pretraining and fine-tuning. Pretraining enables the model to learn denoised and informative entity representations z_i injected with geo-context by solving Eq. (5). Finetuning achieves the specific GER task. Specifically, we base the encoder on BERT, which is used to extract initial entity representations h_i with Eq. (2); to fully consider the neighbor information for token-level denoising, a novel unaligned message passing technique is subsequently applied.

Encoder Architecture Using UMP

Our encoder consists of 3 components (Fig. 2(a)): (1) BERT sequentially embeds textual attributes. (2) UMP layer refines the former entity embeddings. (3) Self-attention layer enables the communication between refined entity tokens.

Unaligned Message-Passing (UMP) To integrate geo-context, it's straightforward to use message-passing to refine entity representations h_i by propagating neighbor features $\{h_j\}_{j \in \mathcal{N}(i)}$. However, existing message-passing layers assume aligned features, typically one-dimensional, while geo-entity from multi-source datasets have different token lengths of $|S(e_i)|$. Though pooling h_i results in an aligned 1-dimension vector, this prevents selective token-level denoising using geo-context (Figure 1). Thus, to adapt message-passing for unaligned node features and enable token-level denoising, we devise a new message computation structure based on cross-attention and rewrite Eq. (3) and Eq. (4),

$$Q_i^{l-1} = h_i^{l-1} W_Q, \quad K_j^{l-1} = h_j^{l-1} W_K, \quad V_j^{l-1} = h_j^{l-1} W_V \tag{6}$$

$$m_{ij}^{l-1} = softmax\left(\frac{Q_i^{l-1} {K_j^{l-1}}^T}{\sqrt{d_k}}\right) V_j^{l-1} \tag{7}$$

$$h_i^l = \mathrm{UPDATE}^l(h_i^{l-1}, \mathrm{AGG}^l(\{\{(m_{ij}^{l-1}, r_{ij}) : j \in \mathcal{N}(i)\}\})) \tag{8}$$

where h_i^{l-1} and h_i^l represent the embeddings of node n_i before and after the l-th convolution and have the same shape. $W_Q \in \mathbb{R}^{d \times d_q}$, $W_K \in \mathbb{R}^{d \times d_k}$, and $W_V \in \mathbb{R}^{d \times d}$ are trainable parameters ($d_q = d_k$). Message m_{ij}^{l-1} has shape $\mathbb{R}^{|S(e_i)| \times d}$. Unlike existing message-passing layers that struggle with varying token lengths, our approach allows n_i's tokens to query spatial neighbor n_j's misaligned tokens.

Finally, classic graph convolution (Kipf and Welling 2016; Velickovic et al. 2017) theory can be applied to AGG and UPDATE. We base the UMP on the Graph Attention Network (Velickovic et al. 2017) for its effectiveness and scalability. As we only consider information propagation from neighbors to a pivot entity, one layer is sufficient. The single-layer UMP in our experiments is detailed as:

$$s = avg(h) \cdot W_1 \tag{9}$$

$$\alpha_{ij} = \sigma(LeakyReLU([s_i; s_j] \cdot W_2) + r_{ij} \cdot W_3) \tag{10}$$

$$\mathrm{UMP}(h_i) = h_i + \sum \alpha_{ij} \cdot m_{ij} \tag{11}$$

where $avg(\cdot)$ averages refined tokens to obtain a coarsened representation $s \in \mathbb{R}^d$. [;] signifies concatenation; $r_{ij} \in \mathbb{R}^1$ represents the Haversine distance between entities i and j; σ is an activation function mapping values between 0 and 1; α_{ij} is the normalized attention score; $W_1 \in \mathbb{R}^{d \times d'}$, $W_2 \in \mathbb{R}^{2d' \times 1}$, $W_3 \in \mathbb{R}^{1 \times 1}$ are trainable parameters.

Contextualized-Pretraining (CP)

We view the embeddings sequence output by our encoder as z and increase a lower bound for Eq. (5) with two tasks. During pretraining, we empirically retrieve neighbors with a random cutoff: we randomly select 50 to 150 of the nearest neighbors within 1000 meters of the pivot entity.

Task #1: Adjacent Entity Prediction (AEP). Intuitively Spatial distance is proportional to the correlation between geo-entities, supported by *Tobler's First Law of Geography: "Everything is related to everything else, but near things are more related than distant things"* (Tobler 1970). To capture this relation, we devise a pre-training task of binarized adjacent entity prediction, easily generated from any multi-source geo-entity database. Specifically, each data sample consists of 2 entities from different databases, denoted as (A, B). (A, B) is labeled as IsAdjacent when selected B is the closest entity to A, and (A, B) is labeled as NotAdjacent if B is a randomly selected entity. We then construct the pseudo-sentences $S(A)$ for A and $S(B)$ for B based on Eq. (1), and combine them into a new sentence for pre-training:

$$Comb(A, B) = \text{[CLS]}\ S(A)\ \text{[SEP]}\ S(B) \quad (12)$$

where [CLS] and [SEP] are special tokens in BERT. The BERT encodes $Comb(A, B)$ to yield word embeddings E:

$$[E_{cls}]\ [E_A^1\ \ldots\ E_A^N]\ [E_{sep}]\ [E_B^1\ \ldots\ E_B^M] \quad (13)$$

We separate $[E_A^1\ \ldots\ E_A^N]$ as h_A, and $[E_B^1\ \ldots\ E_B^M]$ as h_B, and refine these two representations using their respective neighbors through the same UMP layer. Finally, we apply the self-attention mechanism to update the embedding of the CLS token, where its output embedding will be used for adjacent entity prediction (Figure 2(a)). Notably, when generating embeddings for textual attributes, coordinates are excluded, given that the pre-trained labels are based on relationships defined by coordinates. We aim to recover the spatial relationships by comparing the neighboring entities.

Task #2: Masked Language Modeling (MLM). Inspired by BERT, we apply MLM to pseudo-sentence, motivating the model to recover masked tokens utilizing linguistic and geo-contexts from its neighbors. In the dataset generated for Task #1, we randomly select 30% of the attribute/value token positions in pseudo-sentence for prediction. If the i-th token is chosen, it is replaced by the [MASK] token 50% of the time, and by a random token the other 50% of the time.

Theoretical Analysis. Eq. (5) can be maximized based on the Lagrange multiplier, constraints are satisfied for $\lambda \to \infty$, we have: $\max_{z_i}(1+\lambda)I(x_i; z_i) - \lambda I(z_i)$. CP aims to maximize $(1+\lambda)I(x_i; z_i)$. For MLM, given pivot entity e_i and its neighbors $\mathcal{N}(i)$, entity set $\{e_i\} \cup \mathcal{N}(i)$ is randomly masked to obtain $\{e_i'\} \cup \mathcal{N}'(i)$, which is then encoded by a neighbor-aware encoder f to produce the masked representation as $z_i = f(e_i' \cup \mathcal{N}'(i))$. We approximate the maximization of $I(e_i; z_i)$ by reconstructing the tokens of e_i from z_i. If e_i for CP is a low error observation of x_i (i.e., $I(x_i; e_i) \approx I(e_i)$), we have a lower bound of $I(x_i; z_i) \geq I(e_i; z_i)$.

Proof. Chain rule gives two decompositions of $I(z_i; x_i, e_i)$:

$$I(z_i; x_i) + I(z_i; e_i|x_i) = I(z_i; e_i) + I(z_i; x_i|e_i) \quad (14)$$

Based on $I(x_i; e_i) \approx I(e_i)$, we know that e_i is completely determined by x_i, which means the conditional mutual information $I(z_i; e_i|x_i) = 0$. $I(x_i; z_i) \geq I(e_i; z_i)$ follows from the non-negativity of $I(z_i; x_i|e_i)$ and symmetry of $I(;)$.

Yet, MLM may trivially lead to $I(x_i; z_i) \to I(e_i)$. To make $I(x_i; z_i) \to I(x_i; e_i \cup \mathcal{N}(i))$, we view AEP as matching "regions". Since pre-training does not encode coordinates, this motivates z_i to integrate more neighbor information. During fine-tuning, we reduce $\lambda I(z_i)$ and retain factual part by projecting z_i onto a lower-dimensional vector.

Finetuning on GER

We follow the prevailing Blocking then Matching pipeline.

Blocking. Due to GER's originally quadratic time complexity of $|D_1| \times |D_2|$ comparisons, Blocking reduces the computational cost to the most likely matches denoted by $\mathcal{C}$. We adopt the strategy in (Balsebre et al. 2022), selecting candidate pairs based on textual similarities and spatial distance,

$$sim(name_{e_i}, name_{e_j}) \geq 0.6 \wedge dist(e_i, e_j) \leq 1000 \quad (15)$$

where $sim(\cdot)$ and $dist(\cdot)$ are measured by Levenshtein Edit distance and Haversine distance, respectively. Experimentally, we choose the distance threshold based on the baseline's blocking settings to maximize recall.

Matching. We use our pre-trained encoder for matching in $\mathcal{C}$, integrating it with BERT-based matchers (Balsebre et al. 2022; Li et al. 2020). Note that our encoder requires additional neighbors for injecting geo-context; unless otherwise specified, we select neighbors with Eq. (15) by default.

Experiments

Experimental results are reported as the F1 score on the test set, selected from the epoch with the highest validation F1. The number of algorithm runs for each reported result is 10.

Datasets

Our experiments use three renowned geo-entity databases: OpenStreetMap (OSM) is a collaborative open-source mapping project with points of interest such as landmarks; Yelp and Foursquare (FSQ) provide user-generated content, offering insights into business, urban mobility, social dynamics.

Pretraining Dataset. We use data from Singapore, Edinburgh, Toronto, and Pittsburgh, combining OSM with Yelp to create OSM-Yelp and OSM with FSQ to create OSM-FSQ data source. For each city, we generate 5,000 positive sample pairs by selecting entity A from either OSM-Yelp or OSM-FSQ and the nearest distinct entity B from the other data source. An equal number of negative pairs is created by randomly pairing entities from different positive pairs.

Finetuning Dataset. Previous work (Balsebre et al. 2022) provides a GER dataset with more than ten thousand labeled pairs, including 4 cities. The annotated data is divided into training, validation, and test sets, as detailed in Table 1.

Baselines

We compare work with relevant baselines from previous work, including Geo-Entity-Based methods, such as Geo-ER (Balsebre et al. 2022), GeoVectors (Tempelmeier, Gottschalk, and Demidova 2021), SpaBERT (Li et al. 2022), GeoLM (Li et al. 2023); Generic-Entity-Based methods like DeepMatcher (Mudgal et al. 2018), Ditto (Li et al. 2020). The variants of Ditto using BERT (Devlin et al. 2019), ALBERT (Lan et al. 2019), RoBERTa (Liu et al. 2019), DistilBERT (Sanh et al. 2019) are also considered. For all baselines, we use the code provided by the authors and retain the best results within their original parameter options. GeoVectors, SpaBERT, and GeoLM are encoders for embedding geo-entity; For matching, we use an MLP to classify the concatenated embedding of candidate pair, tuning parameters such as the number of layers and hidden dimensions.

Split	Source	Singapore	Edinburgh	Toronto	Pittsburgh
All-Available	OSM/FSQ/Yelp	23,985/31,936/13,699	11,389/7,549/3,868	38,286/18,851/17,204	9,387/11,579/6,356
FT-Train/Valid/Test	OSM-FSQ	1,000/454/2,778	1,000/454/5,246	1,000/454/6,270	1,000/454/1,454
	OSM-Yelp	1,000/454/4,428	1,000/454/3,166	1,000/454/9,398	1,000/454/1,790

Table 1: Number of available entities (All-*) and annotated pairs for finetune (FT-*).

	OSM-FSQ				OSM-Yelp			
	Singapore	Edinburgh	Toronto	Pittsburgh	Singapore	Edinburgh	Toronto	Pittsburgh
20% perturbation ratio								
DeepMatcher	.720±.016	.726±.009	.712±.007	.726±.006	.663±.023	.774±.006	.730±.002	.752±.014
Ditto-BERT (0.05)	.703±.004	.758±.008	.728±.005	.729±.008	.738±.004	.816±.015	.763±.009	.787±.012
GeoVector	.601±.022	.529±.018	.568±.199	.539±.023	.491±.198	.667±.004	.679±.001	.663±.001
SpaBERT	.644±.004	.650±.009	.646±.010	.689±.005	.412±.275	.683±.004	.684±.007	.671±.009
GeoLM	.679±.010	.643±.007	.696±.013	.701±.004	.663±.012	.660±.005	.726±.011	.733±.005
Geo-ER (0.48)	*.725±.013*	*.818±.002*	*.797±.004*	*.807±.004*	*.800±.002*	*.883±.003*	*.847±.007*	*.853±.001*
Geo-ER+**UMP&CP** (1)	.802±.008	.836±.006	.822±.007	**.836±.008**	**.830±.005**	**.908±.001**	**.870±.004**	**.890±.001**
BERT+**UMP&CP** (0.58)	**.804±.004**	**.836±.001**	**.826±.004**	.834±.005	.829±.002	.905±.003	.865±.004	.884±.003
Improvement	<u>7.9%</u>	<u>1.8%</u>	<u>2.9%</u>	<u>2.9%</u>	<u>3.0%</u>	<u>2.5%</u>	<u>2.3%</u>	<u>3.7%</u>
40% perturbation ratio								
DeepMatcher	*.680±.005*	.706±.002	.670±.019	.687±.007	.662±.001	.729±.006	.687±.009	.724±.014
Ditto-BERT (0.05)	.675±.006	.691±.006	.690±.004	.689±.008	*.687±.009*	.724±.008	.682±.017	.721±.008
GeoVector	.583±.017	.526±.049	.529±.159	.505±.026	.446±.067	.665±.005	.671±.009	.652±.061
SpaBERT	.636±.002	.630±.003	.633±.006	.680±.012	.512±.006	.662±.009	.669±.003	.655±.009
GeoLM	.665±.006	.639±.004	.674±.010	.693±.007	.643±.005	.671±.017	.689±.003	.719±.013
Geo-ER (0.48)	.618±.030	*.742±.005*	*.698±.016*	*.720±.006*	.672±.011	*.775±.011*	*.752±.003*	*.769±.007*
Geo-ER+**UMP&CP** (1)	**.740±.007**	**.768±.001**	.758±.010	**.760±.007**	**.762±.001**	.817±.004	**.807±.008**	**.833±.005**
BERT+**UMP&CP** (0.58)	.739±.011	.756±.009	**.758±.002**	.755±.003	.756±.004	**.823±.003**	.804±.007	.825±.003
Improvement	<u>6.0%</u>	<u>2.6%</u>	<u>6.0%</u>	<u>4.0%</u>	<u>7.5%</u>	<u>4.8%</u>	<u>5.5%</u>	<u>6.4%</u>

Table 2: F1 under 20%/40% perturbation ratio; **highest**, <u>significant</u>, *best alternative* results and (norm time-cost) are highlighted.

Implementation Details

We integrate our proposal (denoted by +*UMP&CP*) into the state-of-the-art method Geo-ER and a foundational LM BERT without altering their hyperparameters. For all additional attention layers introduced, the dimensions of the query and key are set to 256 (corresponding to d_q and d_k in Eq. (6)); the dimension of the value $d_v = d$ matches the output feature dimension of the expanded transformer; in Eq. (11), dimension for aggregation d' is set to 256, and the activation function σ is set to be sigmoid or softmax.

Perturbation-Based Evaluation

Our research addresses geo-entity resolution (GER) under poor dataset quality conditions, such as spelling errors and missing values. Since no geospatial dataset includes standard metrics for data quality, we introduce a perturbation ratio to quantitatively analyze method robustness. This ensures experiments are conducted under controlled poor-quality conditions, reducing data variations by comparing performance before and after perturbations. Given a perturbation ratio ρ, we randomly select ρ% of attribute/value token positions in *all entities* for perturbation, replacing tokens with either [MASK] (50%) or random tokens (50%), to simulate value missing and spelling error respectively. Latitude and longitude from geo-positional systems are excluded from perturbation. We then use these perturbed entities as input for various GER methods. For a fair comparison, all neighbor-based methods use the same neighbor selection scheme based on Eq. (15), followed by perturbations.

Results

Results on General Perturbation. Table 2 details the experimental results under the perturbation ratio of $\rho = 20\%$ and $\rho = 40\%$, a common span that is most likely to be encountered in poor data quality scenarios. On each dataset, the algorithm with the highest F1 is highlighted in bold. For clarity, the last row shows the improvement of our proposal over the best alternative; the significant improvement is underlined (confidence level is 95%). Our method significantly improves the performance of the baseline method on all datasets, achieving the highest F1 and exceeding the best alternatives by at least 1.8% to a maximum of 7.9%. Based on these results, we also have several findings.

First, when the data quality is slightly perturbed ($\rho = 20\%$), methods that learn neighbor information in a graph manner (Geo-ER and our proposals) consistently outperform those that learn in a sequence manner (SpaBERT, GeoLM). Interestingly, sequence-based methods do not even surpass some general ER methods that do not utilize neighbor information (Ditto, DeepMatcher), indicating that the way neighbor information is utilized matters. Second, as the perturbation ratio rises from 20% to 40%, w.r.t noise in the dataset increases, the performance of Geo-ER on the Singapore dataset becomes steep and also falls behind most of the general ER methods, due to Geo-ER not denoising tokens

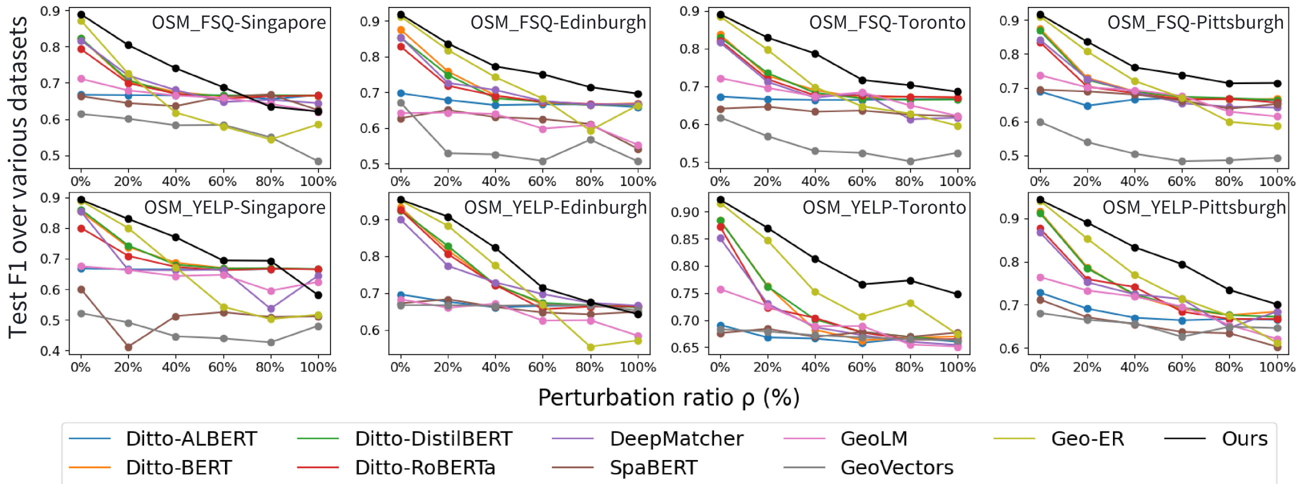

Figure 3: Impact of different perturbation ratio ρ on robustness evaluation.

	OSM-FSQ				OSM-Yelp			
	Singapore	Edinburgh	Toronto	Pittsburgh	Singapore	Edinburgh	Toronto	Pittsburgh
Geo-ER + **UMP**&**CP**	$.802 \pm .008$	$\mathbf{.836 \pm .006}$	$.822 \pm .007$	$\mathbf{.836 \pm .008}$	$\mathbf{.830 \pm .005}$	$\mathbf{.908 \pm .001}$	$.870 \pm .004$	$\mathbf{.890 \pm .001}$
(-) **UMP**	$\mathbf{.805 \pm .001}$	$.833 \pm .001$	$.817 \pm .005$	$.824 \pm .010$	$.824 \pm .006$	$.901 \pm .004$	$.850 \pm .003$	$.872 \pm .005$
(-) **CP** (**MLM**&**AEP**)	$.780 \pm .007$	$.830 \pm .005$	$.816 \pm .007$	$.820 \pm .015$	$.807 \pm .004$	$.898 \pm .002$	$.862 \pm .002$	$.884 \pm .005$
(-) **MLM**	$.777 \pm .001$	$.824 \pm .003$	$.814 \pm .003$	$.811 \pm .004$	$.810 \pm .011$	$.894 \pm .009$	$.851 \pm .002$	$.877 \pm .006$
(-) **AEP**	$.792 \pm .005$	$.810 \pm .002$	$.816 \pm .005$	$.823 \pm .004$	$.812 \pm .011$	$.893 \pm .005$	$.844 \pm .004$	$.881 \pm .004$
(+) Trivial Neighbors	$.796 \pm .002$	$.826 \pm .011$	$\mathbf{.825 \pm .006}$	$.830 \pm .006$	$.813 \pm .003$	$.901 \pm .004$	$\mathbf{.872 \pm .007}$	$.884 \pm .004$
Geo-ER	$.725 \pm .013$	$.818 \pm .002$	$.797 \pm .004$	$.807 \pm .004$	$.800 \pm .002$	$.883 \pm .003$	$.847 \pm .007$	$.853 \pm .001$
(+) Trivial Neighbors	$.720 \pm .007$	$.813 \pm .006$	$.798 \pm .004$	$.806 \pm .006$	$.787 \pm .003$	$.878 \pm .005$	$.836 \pm .002$	$.849 \pm .007$

Table 3: Ablation results under a 20% perturbation ratio, highest F1 is highlighted in bold, (-) indicates the removed component.

when injecting neighbors, thus introducing more noise; in contrast, our method is capable of selectively capturing fine-grained neighbor information and can maintain top-tier performance on noisy data. Lastly, we observe that the basic adaptation (BERT+UMP&CP) also yields competitive results and shows an average 9% improvement over its original application (Ditto-BERT), highlighting our generalizability.

Impact of Different Perturbation. The full results in Figure 3 are conducive to a comprehensive understanding of the robustness experiments. As ρ varies from 0% to 100%, the quality of the dataset becomes increasingly poor. We note that $\rho = 0$ indicates no perturbation, which is a standard benchmark widely adopted in previous work (Balsebre et al. 2022); in this setting, various methods can achieve competitive performance because comparing the semantics of two entities can essentially distinguish them, and the introduction of geo-context injection in our method can only bring a slight improvement. For the case when $\rho \leq$60%, our proposal consistently ranks top-1 over all datasets, and its performance degrades more slowly compared to alternatives as the perturbation ratio increases; this indicates that leveraging geo-context can enhance the robustness of GER. It is interesting to note that when $\rho \geq 80\%$, our method's performance is no longer consistently dominant; we attribute the decline to the extremely poor data quality, where neighbors hardly provide useful information and even contribute more noise; yet we still exceed the baselines that utilize neighbors and is competitive to the general ER methods.

Ablation Results. To justify our design, we remove UMP and CP individually to assess their impact. As shown in Table 3, both UMP and CP enhance baseline performance independently, and their combination results in stable performance. Removing tasks such as AEP or MLM leads to worse results than removing CP, highlighting the importance of learning both fine-grained tokens (MLM) and coarse-grained regions (AEP). We also test the impact of neighbor selection by replacing spatially and textually similar neighbors (Eq. (15)) with *Trivial Neighbors* (the 50 closest within the same distance block). This results in a decline in Geo-ER's average performance, with no improvement in rank. However, our method still maintains top-tier performance even with trivial neighbors, validating our proposal.

Conclusion

To tackle the real-world challenge of robust geo-entity resolution, we introduce a new message-passing module (UMP) and a contextualized-pretraining framework (CP) that efficiently leverages unaligned attributes across spatial neighbors for geo-context injection. Extensive experiments and ablation studies demonstrate our superiority in terms of robustness. *The code and processed data can be found in this url:* https://github.com/2022neo/ger_ump_cp

Acknowledgments

This research is supported by Amap. Yuwen Ji is supported by the National Key R&D Program of China (2021YFB3500700), NSFC Grant 62172026, National Social Science Fund of China 22&ZD153, the Fundamental Research Funds for the Central Universities, State Key Laboratory of Complex Critical Software Environment (SKLCCSE), Lei Shi is his supervisor.

References

Abdalla, R. 2016. Geospatial Data Integration.

Balley, S.; Parent, C.; and Spaccapietra, S. 2004. Modeling geographic data with multiple representations. *International Journal of Geographic Information Systems*, 329–354.

Balsebre, P.; Huang, W.; Cong, G.; and Li, Y. 2023a. City Foundation Models for Learning General Purpose Representations from OpenStreetMap.

Balsebre, P.; Yao, D.; Cong, G.; and Hai, Z. 2022. Geospatial Entity Resolution. *Proceedings of the ACM Web Conference 2022*.

Balsebre, P.; Yao, D.; Cong, G.; Huang, W.; and Hai, Z. 2023b. Mining Geospatial Relationships from Text. *Proceedings of the ACM on Management of Data*, 1: 1 – 26.

Barlaug, N.; and Gulla, J. A. 2020. Neural Networks for Entity Matching: A Survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15: 1 – 37.

Bilenko, M.; and Mooney, R. J. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Knowledge Discovery and Data Mining*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T. J.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *ArXiv*, abs/2005.14165.

Christophides, V.; Efthymiou, V.; Palpanas, T.; Papadakis, G.; and Stefanidis, K. 2019. End-to-End Entity Resolution for Big Data: A Survey. *ArXiv*, abs/1905.06397.

Dalvi, N. N.; Rastogi, V.; Dasgupta, A.; Sarma, A. D.; and Sarlós, T. 2013. Optimal hashing schemes for entity matching. *Proceedings of the 22nd international conference on World Wide Web*.

Deng, Y.; Luo, A.; Liu, J.; and Wang, Y. 2019. Point of Interest Matching between Different Geospatial Datasets. *ISPRS Int. J. Geo Inf.*, 8: 435.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics*.

Dwivedi, V. P.; Joshi, C. K.; Luu, A. T.; Laurent, T.; Bengio, Y.; and Bresson, X. 2020. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*.

Ebraheem, M.; Thirumuruganathan, S.; Joty, S. R.; Ouzzani, M.; and Tang, N. 2017a. DeepER - Deep Entity Resolution. *ArXiv*, abs/1710.00597.

Ebraheem, M.; Thirumuruganathan, S.; Joty, S. R.; Ouzzani, M.; and Tang, N. 2017b. Distributed Representations of Tuples for Entity Resolution. *Proc. VLDB Endow.*, 11: 1454–1467.

Elmagarmid, A. K.; Ilyas, I. F.; Ouzzani, M.; Quiané-Ruiz, J.-A.; Tang, N.; and Yin, S. 2014. NADEEF/ER: generic and interactive entity resolution. *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*.

Firmani, D.; Saha, B.; and Srivastava, D. 2016. Online Entity Resolution Using an Oracle. *Proc. VLDB Endow.*, 9: 384–395.

Fu, C.; Han, X.; He, J.; and Sun, L. 2020. Hierarchical Matching Network for Heterogeneous Entity Resolution. In *International Joint Conference on Artificial Intelligence*.

Fu, C.; Han, X.; Sun, L.; Chen, B.; Zhang, W.; Wu, S.; and Kong, H. 2019. End-to-End Multi-Perspective Matching for Entity Resolution. In *International Joint Conference on Artificial Intelligence*.

Ge, C.; Wang, P.; Chen, L.; Liu, X.; Zheng, B.; and Gao, Y. 2023. CollaborEM: A Self-Supervised Entity Matching Framework Using Multi-Features Collaboration. *IEEE Transactions on Knowledge and Data Engineering*, 35: 12139–12152.

Gokhale, C. S.; Das, S.; Doan, A.; Naughton, J. F.; Rampalli, N.; Shavlik, J. W.; and Zhu, X. 2014. Corleone: hands-off crowdsourcing for entity matching. *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*.

Isaj, S.; Zimányi, E.; and Pedersen, T. 2019. Multi-Source Spatial Entity Linkage. *IEEE Transactions on Knowledge and Data Engineering*, 34: 1344–1358.

Kipf, T.; and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *ArXiv*, abs/1609.02907.

Konda, P.; Das, S.; PaulSuganthanG., C.; Doan, A.; Ardalan, A.; Ballard, J. R.; Li, H.; Panahi, F.; Zhang, H.; Naughton, J. F.; Prasad, S.; Krishnan, G.; Deep, R.; and Raghavendra, V. 2018. Technical Perspective:: Toward Building Entity Matching Management Systems. *SIGMOD Rec.*, 47: 33–40.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *ArXiv*, abs/1909.11942.

Lang, C.; Cheng, G.; Tu, B.; Li, C.; and Han, J. 2023. Retain and Recover: Delving Into Information Loss for Few-Shot Segmentation. *IEEE Transactions on Image Processing*, 32: 5353–5365.

Li, B.; Miao, Y.; Wang, Y.; Sun, Y.; and Wang, W. 2021. Improving the Efficiency and Effectiveness for BERT-based Entity Resolution. In *AAAI Conference on Artificial Intelligence*.

Li, Y.; Li, J.; Suhara, Y.; Doan, A.; and Tan, W. C. 2020. Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment*, 14: 50 – 60.

Li, Z.; Kim, J.; Chiang, Y.-Y.; and Chen, M. 2022. SpaBERT: A Pretrained Language Model from Geographic Data for Geo-Entity Representation. *ArXiv*, abs/2210.12213.

Li, Z.; Zhou, W.; Chiang, Y.-Y.; and Chen, M. 2023. GeoLM: Empowering Language Models for Geospatially Grounded Language Understanding. In *Conference on Empirical Methods in Natural Language Processing*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, abs/1907.11692.

Miller, G. A.; Beckwith, R.; Fellbaum, C. D.; Gross, D.; and Miller, K. J. 1990. Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3: 235–244.

Morana, A.; Morel, T.; Berjawi, B.; and Duchateau, F. 2014. GeoBench: a geospatial integration tool for building a spatial entity matching benchmark. *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*.

Mudgal, S.; Li, H.; Rekatsinas, T.; Doan, A.; Park, Y.; Krishnan, G.; Deep, R.; Arcaute, E.; and Raghavendra, V. 2018. Deep Learning for Entity Matching: A Design Space Exploration. *Proceedings of the 2018 International Conference on Management of Data*.

Nie, H.; Han, X.; He, B.; Sun, L.; Chen, B.; Zhang, W.; Wu, S.; and Kong, H. 2019. Deep Sequence-to-Sequence Entity Matching for Heterogeneous Entity Resolution. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

OpenAI. 2023. GPT-4 Technical Report. https://api.semanticscholar.org/CorpusID:257532815. Accessed: 2024-06-17.

Peeters, R.; and Bizer, C. 2021. Dual-Objective Fine-Tuning of BERT for Entity Matching. *Proc. VLDB Endow.*, 14: 1913–1921.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Sarawagi, S.; and Bhamidipaty, A. 2002. Interactive deduplication using active learning. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Schäfers, M.; and Lipeck, U. W. 2014. SimMatching: adaptable road network matching for efficient and scalable spatial data integration. *Proceedings of the 1st ACM SIGSPATIAL PhD Workshop*.

Sehgal, V.; Getoor, L.; and Viechnicki, P. 2006. Entity resolution in geospatial data integration. 83–90.

Shivaprabhu, V. R.; Balasubramani, B. S.; and Cruz, I. F. 2017. Ontology-based Instance Matching for Geospatial Urban Data Integration. *Proceedings of the 3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics*.

Stergiou, A.; and Poppe, R. 2021. AdaPool: Exponential Adaptive Pooling for Information-Retaining Downsampling. *IEEE Transactions on Image Processing*, 32: 251–266.

Tabarro, P. G.; Pouliot, J.; Fortier, R.; and Losier, L.-M. 2017. a Webgis to Support Gpr 3d Data Acquisition: a First Step for the Integration of Underground Utility Networks in 3d City Models. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43–48.

Tempelmeier, N.; Gottschalk, S.; and Demidova, E. 2021. GeoVectors: A Linked Open Corpus of OpenStreetMap Embeddings on World Scale. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

Tobler, W. 1970. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, 46: 234–240.

Vaswani, A.; Shazeer, N. M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Neural Information Processing Systems*.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio', P.; and Bengio, Y. 2017. Graph Attention Networks. *ArXiv*, abs/1710.10903.

Wang, J.; Kraska, T.; Franklin, M. J.; and Feng, J. 2012. CrowdER: Crowdsourcing Entity Resolution. *Proc. VLDB Endow.*, 5: 1483–1494.

Wang, J.; Li, G.; Yu, J. X.; and Feng, J. 2011. Entity Matching: How Similar Is Similar. *Proc. VLDB Endow.*, 4: 622–633.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.