Geospatial Entity Resolution

Pasquale Balsebre
Nanyang Technological University
Singapore
pasquale001@e.ntu.edu.sg

Dezhong Yao
Huazhong University of Science and Technology
China
dyao@hust.edu.cn

Gao Cong
Nanyang Technological University
Singapore
gaocong@ntu.edu.sg

Zhen Hai
DAMO Academy, Alibaba Group
Singapore
haiz0001@e.ntu.edu.sg

ABSTRACT

A geospatial database is today at the core of an ever increasing number of services. Building and maintaining it remains challenging due to the need to merge information from multiple providers. Entity Resolution (ER) consists of finding entity mentions from different sources that refer to the same real world entity. In geospatial ER, entities are often represented using different schemes and are subject to incomplete information and inaccurate location, making ER and deduplication daunting tasks. While tremendous advances have been made in traditional entity resolution and natural language processing, geospatial data integration approaches still heavily rely on static similarity measures and human-designed rules. In order to achieve automatic linking of geospatial data, a unified representation of entities with heterogeneous attributes and their geographical context, is needed. To this end, we propose Geo-ER[1], a joint framework that combines Transformer-based language models, that have been successfully applied in ER, with a novel learning-based architecture to represent the geospatial character of the entity. Different from existing solutions, Geo-ER does not rely on pre-defined rules and is able to capture information from surrounding entities in order to make context-based, accurate predictions. Extensive experiments on eight real world datasets demonstrate the effectiveness of our solution over state-of-the-art methods. Moreover, Geo-ER proves to be robust in settings where there is no available training data for a specific city.

CCS CONCEPTS

• **Information systems** → *Entity resolution*; • **Geographic information systems** → *Information integration.*

KEYWORDS

Entity resolution, neural networks, geospatial data, neighbourhood embedding, graph attention

ACM Reference Format:
Pasquale Balsebre, Dezhong Yao, Gao Cong, and Zhen Hai. 2022. Geospatial Entity Resolution. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485447.3512026

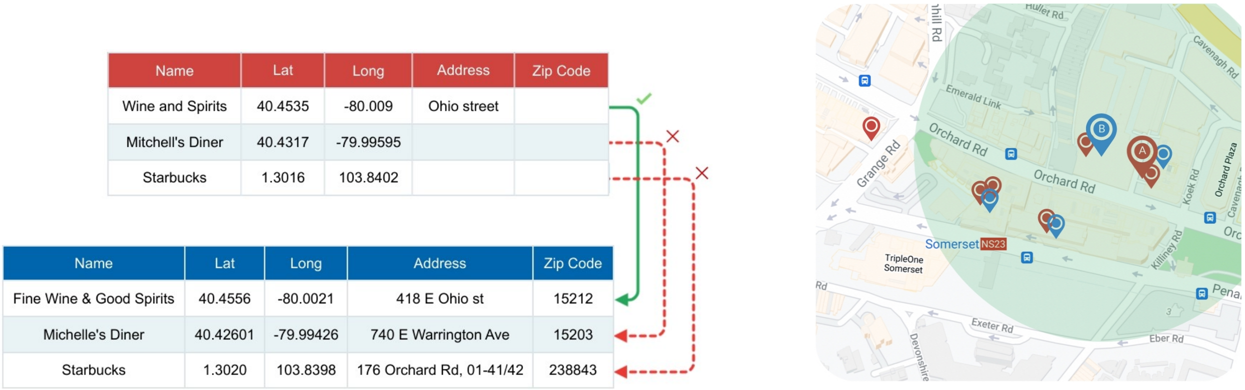
1 INTRODUCTION

A complete and high quality geospatial database is a key element to improve quality of service for navigation, social networks, advertising, and logistics. However, such data is rarely contained in a single source, but rather distributed across several different location-aware applications, like Location-Based Services (LBS) or Map Services, with each provider having only partial coverage of the geographical picture. As a result, there is a great interest in joining information to build a comprehensive overview of geospatial entities. Integrating spatial data from multiple sources poses several challenges: each source, in fact, represents entities with a different schema and data suffers from inconsistency, redundancy, and ambiguity. A suitable linking schema is highly desirable to perform deduplication and considerably enhance the combined database. This represents a critical step towards automatic geospatial integration.

Entity resolution (ER) [4, 6, 12] is the task of finding entity mentions from two different data sources, that refer to the same real-world entity. Given its substantial number of applications, it has received significant attention in recent years. In this work, we focus on geospatial ER. A geospatial entity is typically identified by the combination of a spatial position and a set of textual attributes. An example of the challenges presented by geospatial ER is depicted in Figure 1a. In the first candidate pair, different words are used to express the same concept (*& = and*, *street = st*), the Points of Interest (POIs) names have a different number of words and the address information in the left table is incomplete. However, they are located at a very short distance. All the aforementioned insights must play a role in the final matching decision. In the second example, only the name and the geospatial location are available for comparison: a small difference in the name of the business (*Mitchell's ≠ Michelle's*) is an important clue that the two POIs cannot match. Static similarity measures, applied on the *Name* attribute, would result in a high similarity score, ignoring the role that each word has in the context. Language modeling capabilities are necessary to capture the semantics of the words and figure out a common word like *Diner* in the name, is not informative enough to match the two candidates. Finally, in the last pair, the names perfectly overlap and the distance is still acceptable, considering the error

[1] https://github.com/PasqualeTurin/Geo-ER

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512026



Name	Lat	Long	Address	Zip Code
Wine and Spirits	40.4535	-80.009	Ohio street	
Mitchell's Diner	40.4317	-79.99595		
Starbucks	1.3016	103.8402		

Name	Lat	Long	Address	Zip Code
Fine Wine & Good Spirits	40.4556	-80.0021	418 E Ohio st	15212
Michelle's Diner	40.42601	-79.99426	740 E Warrington Ave	15203
Starbucks	1.3020	103.8398	176 Orchard Rd, 01-41/42	238843

(a) Examples of geospatial entity resolution

(b) Observing the neighbourhood, Geo-ER can capture the context where the entities are located. A, B and the neighbours (smaller pins) are *Starbucks* POIs

Figure 1

in Geo-positional systems; nonetheless *Starbucks* is a very popular chain and the two POIs are located in the dense district of Orchard (Figure 1b), in Singapore: these suggest that it is still reasonable to have two distinct POIs with the same name even if they are in a short distance.

In the past few years, tremendous progress has been made in entity resolution, but most of the existing solutions focus on attribute comparison and language understanding to match *non-spatial* entities [13, 20, 21, 25]. Recent efforts in the geospatial data integration community rely on manually-designed rules [19] or static string similarity measures [24, 40]. In order to address the challenges presented in Figure 1, especially in case of incomplete address information, language modeling techniques are not sufficient and we argue that additional knowledge, like the context where an entity is located, needs to be considered.

We present Geo-ER, a geospatial linking model, with language understanding, numeracy, and spatial capabilities. We design a novel geospatial attention component to make *context-based* predictions. Figure 1b shows how, taking into account the neighbourhood of the entities in the third example of Figure 1a, can *contextualize* their representation, simplifying geospatial ER. Several fields have greatly benefited from contextual information: in natural language processing, *contextual word embedding* models, like ELMo [29] and BERT [9], have obtained state-of-the-art results in several tasks; in computer vision, *context-based vision* simplifies object recognition [34]. Following a similar intuition, we propose to leverage context information for the geospatial linking task. The contributions of this paper can be summarized as follows:

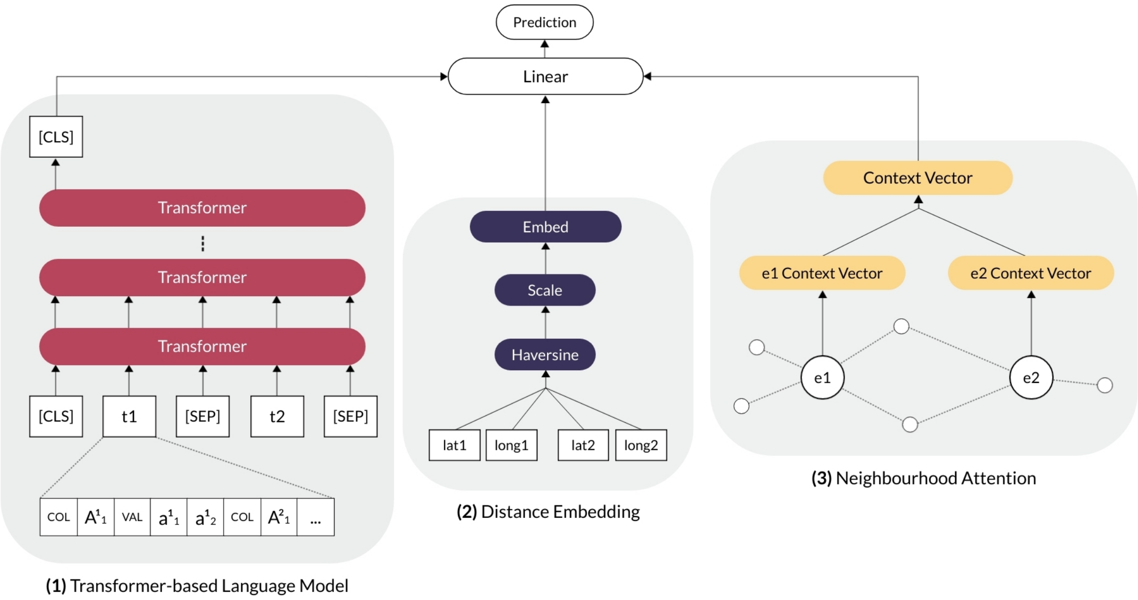
- We introduce Geo-ER, a unified framework to match geospatial entities from different data sources, using their textual attributes, geospatial information, and the context where they are located. This represents a step towards automatic geospatial integration, with various applications ranging from recommendation systems to logistics services.
- We make Geo-ER *context-aware* by developing a neighbourhood embedding component, based on the graph attention (GAT) mechanism [36], together with pre-trained language models for textual attribute comparison. This enables Geo-ER to produce a more contextualized representation of the entity, resulting in higher accuracy in ER. No previous work has proposed using surrounding entities' information to improve geospatial data integration.
- Extensive experiments on eight real-world datasets, from three different sources, are performed along with comparisons to existing state-of-the-art algorithms, including the recent approach Ditto [21], to demonstrate the advantages of our proposed solution. Geo-ER can rely on both textual and geospatial components and displays robustness with respect to missing attributes, imprecise positional information, and cross-city validation.

2 RELATED WORK

2.1 Entity Resolution

Several proposals have tackled the matching problem in Entity Resolution using human-designed rules [7, 11, 39] and crowd-sourcing [16, 38]. Rule-based solutions benefit from the high interpretability of the models but are time- and resource-expensive, requiring the involvement of domain experts and performing poorly on *non-structured* data [25]. Deep Learning-based approaches have emerged as the state-of-the-art solutions for ER, mostly due to their ability to dynamically learn a distributed representation of the entities. DeepER [10] and DeepMatcher [25] are two pioneering solutions in the adoption of deep learning architectures for ER. DeepER uses LSTMs to represent tuples in a distributed fashion, whereas DeepMatcher proposes a sequence-aware model combining RNNs and attention mechanism. Following the DeepMatcher framework, M2M [15], Seq2Seq [26] and HierMatcher [14] are proposed to get better performance on some datasets.

Recently, pre-trained language models (LMs), such as BERT [9], that have been trained on unsupervised language modeling tasks on massive text corpora have been used for entity resolution and achieved better accuracy. Ditto [21] casts ER as a sequence-pair classification problem based on fine-tuning pre-trained LMs and proposes domain knowledge injection to highlight specific spans of tokens, as well as several data augmentation operators. Li et



(1) Transformer-based Language Model

(2) Distance Embedding

(3) Neighbourhood Attention

Figure 2: A general overview of Geo-ER. The three main components are: (1) Transformer-based language model (Section 3.3) for textual attributes comparison, (2) Distance Embedding component (Section 3.4) to compute and embed the geographical distance between the two entities, (3) Neighbourhood Attention (Section 3.5) to embed information of the surrounding entities. As shown in the figure, in the LM input, *t1* and *t2* are the serialization of the two entities *e1* and *e2*

.

al. [20] make use of a siamese network structure based on BERT, both to speed up the blocking phase and compare candidate pairs. Peeters and Bizer [27] present JointBERT for entity matching with multi-class classification. All the aforementioned solutions are not designed to consider the geographical character of an entity. Ditto supports many pre-trained LMs, its optimization techniques are relevant for the geospatial linking task and it achieved state-of-the-art results on a wide number of ER benchmark datasets, making it the most suitable algorithm for empirical comparison with Geo-ER.

2.2 Geospatial Entity Resolution

Some contributions in geographical data integration focus on purely spatial objects, i.e., entities that are determined solely based on their coordinates. Solutions in [1, 3, 33] aim to create a unified spatial representation for spatial objects deriving from sensors and radars. Schäfers and Lipeck present SimMatching [31] for similarity-based spatial matching of road networks. Comparisons are performed between road attributes (e.g., length, shape, road name) to filter out duplicates in an integrated database. Unlike purely spatial objects, spatial entities are identified by a combination of spatial and textual attributes. Different spatial entities might share the same name (e.g., a chain of restaurants) or the same position (e.g., a shopping mall), making the task of entity resolution more challenging. Existing works tackle the problem of geospatial ER using different rules to compare both spatial and non-spatial attributes. In [8] Deng et al. specify a set of attribute similarity measures, a simple attribute selection strategy and use the improved D-S evidence theory to combine attribute-matched results. Shivaprabu et al. [32] propose an ontology-based instance matching architecture to integrate geospatial urban data. In [24], Morana et al. use Euclidean distance to measure the distance between the POIs, Levenshtein similarity for textual attributes like address and name, Resnik similarity (Wordnet [23]) for the category. In [18] Isaj et al. introduce *SkyEx* to match POIs from different sources, based on the Pareto optimality, without the need of weights, scoring functions, nor a training set and achieve state-of-the-art results in multi-source spatial entity linkage. Most of the aforementioned works still heavily rely on manually-designed rules and thresholds or textual similarity metrics, which fail to capture the semantics of the words. Moreover, none of them takes into account information from surrounding entities during the matching decision.

3 METHOD

We first formalize the geospatial ER problem in Section 3.1 and subsequently present our solution, component by component, justifying our design choices as well as their respective advantages.

3.1 Problem Setting

The problem of geospatial Entity Resolution considered in this work is stated as follows: Given two geospatial databases D_1 and D_2, containing $|D_1|$ and $|D_2|$ records, respectively, we aim to find all the couples of entities, one from D_1 and the other from D_2, that refer to the same real-world entity. Such entities are called *matches.* A geospatial entity e_i is identified by a set of textual attributes $\{name_i$, $address_i$, $zipcode_i\}$ and a geospatial position $\{latitude_i$, $longitude_i\}$. In the ER pipeline, the *matching* phase is always preceded by the *blocking* phase.

Algorithm 1 Geo-ER

```
Require: Geospatial entities datasets $D_1 = \{e^1_1, e^1_2, ...e^1_n\}, D_2 = \{e^2_1, e^2_2, ...e^2_m\}$, Candidate set $C$ from blocking, pre-trained language model (LM)
training;
for each pair $(e_i, e_j)$ in $C$ do
    Retrieve neighbourhood of $(e_i, e_j)$ from blocking
    $\hat{y}_{(e_i,e_j)} = Geo\text{-}ER(LM(e_i, e_j), dist(e_i, e_j), neigh\_attn(e_i, e_j))$
    $CrossEntropyLoss(\hat{y}_{(e_i,e_j)}, y_{(e_i,e_j)})$
    backprop
end for
prediction;
Given unlabeled tuple $t$
Retrieve neighbourhood of $t$ from blocking
result = $Geo\text{-}ER(LM(t), dist(t), neigh\_attn(t))$
return result
```

Blocking. In real-world scenarios, it is infeasible to make $|D_1|{\times}|D_2|$ comparisons, therefore blocking aims to retrieve a candidate set C of entity mention pairs likely to match. In this way, the fine-grained matching functions are applied only on such candidates, thus accelerating the ER pipeline. In the example in Figure 1, after the blocking phase, the total number of comparisons is reduced from nine to three. We design an effective blocking strategy for geospatial data based on both textual similarities and spatial distance. Specifically, the name similarity is measured using the *Levenshtein Edit* distance. Each pair (e_i, e_j) in $D_1 \times D_2$, is deemed to be a candidate pair for matching, if the following holds:

$$Levenshtein(name_{e_i}, name_{e_j}) \geq 0.6 \land dist(e_i, e_j) \leq 2000 \qquad (1)$$

We choose a high distance threshold to maximize the recall. In fact, for many large geospatial entities, like parks or airports, the coordinates are registered in positions far from each other, resulting in a large distance.

Matching. Given a set C of candidate entity mention pairs, we aim to design a *matcher* that can accurately classify each pair as *matching* or *non-matching*. To learn such an algorithm, we use a set T of labeled data. This task falls into the *Dirty ER* category: entity mentions in D_1 and D_2 are structured records, but attributes may be missing or injected under different attributes (e.g., the street name may be part of geospatial entity name).

3.2 Matching Model Overview

The architecture of Geo-ER is summarized in Figure 2. It can be divided into three main components that jointly contribute to the final matching decision: (1) The *Language Model* to compare the textual attributes of the entities, (2) The *Distance Embedding* component to compute and embed the spatial distance between the two geospatial entities, and (3) The *Neighbourhood Attention* to embed information of the surrounding geospatial entities.

3.3 Language Models For Entity Comparison

We design Geo-ER to model the textual and spatial features of an entity using separate components and combine their representations subsequently. To tackle the problem of textual attribute comparison, as shown in the first two examples in Figure 1a, a deep semantic understanding of the words is necessary.

Language models (LMs) based on the Transformer [35] architecture, such as BERT [9] or GPT-2 [30], have established a new state-of-the-art in a variety of NLP tasks [17, 41]. The success of this architecture is largely due to the *self-attention* mechanism. The word embeddings generated by Transformers are deeply-contextualized and capture the different meanings that a word can have in different contexts. Conversely, traditional word embedding techniques like GloVe [28] or FastText [5] would produce the same result regardless of the context. The Entity Resolution community is increasingly adopting LMs for the matching task. Specifically, a pre-trained LM can be fine-tuned on a new task and achieve impressive results. For these reasons, we cast the textual attributes comparison problem as a sequence-pair comparison task, and fine-tune BERT [9], a large LM trained on masked language modeling (MLM) and next sentence prediction (NSP), which best performed in our experiments. Following Ditto [21] we serialize each entity as:

$$\text{Serialize}(e) = \text{[COL]}\ attr_i\ \text{[VAL]}\ value_i\ ...\ \text{[COL]}\ attr_k\ \text{[VAL]}\ value_k$$

where the token [COL] precedes the attribute name and the token [VAL] its value. This schema signals to the model which attribute a set of tokens belongs to and concurrently enables a *schema-agnostic* model. A candidate couple of entities is joined as:

$$\text{Combine}(e_i, e_j) = \text{[CLS] Serialize}(e_i)\ \text{[SEP] Serialize}(e_j)\ \text{[SEP]}$$

where [SEP] is a special token to separate sequences and [CLS] is used to encode the candidate pair into a d_{hidden}-dimensional vector that will contain information on the similarity of the entities. The output of the LM is a sequence of the same length of the input, where each token is a d_{hidden}-dimensional contextualized embedding of the input word. We select only the first one, corresponding to the [CLS] token, and concatenate it to the output of the other components for the final decision. The choice to keep the entities together in the input comes with the advantage of a *cross-entity* and *cross-attribute* comparison, without the need to align and compare their encodings subsequently. This is an important advantage in *Dirty ER*, since information injected under different attributes is automatically aligned and compared, by means of *self-attention* mechanism.

3.4 Distance Embedding

To match spatial entities from different sources, textual attributes like name, address, or postal code may not be enough for an accurate result. In fact, different POIs may share the same name or be in the same building, with a similar or identical address; conversely, the same POI may be accessed by users from different entries, resulting in different addresses. Moreover, as shown in Table 2, only a subset of samples contains address information on both the compared entities. The first candidate pair in Figure 1a, is a typical example in which textual attributes cannot provide sufficient information to correctly classify the pair. The spatial distance between the two candidate entities must be appropriately considered by a geospatial-linking model.

Language models, like BERT, have a fixed-size vocabulary, and use a word-splitting scheme[2] to represent words that are not part of it (e.g. *'embeddings' = 'em', '##bed', '##ding', '##s'*). In this way, each word can be represented, at the very least, as a collection of its individual characters. This approach carries a number of advantages to represent unseen words, but the same does not hold for numbers. In a recent study [37], Wallace and Wang et al. probe numeracy in embeddings and show that sub-word models suffer from the poor word-splitting method: numbers that are very similar in value may have very different sub-word division, leading to a model misinterpretation of the numbers.

To address this problem, we decide to add a separate component for distance calculation and embedding in our architecture. This choice of design enables numeracy in our model and proves to be essential in the experiments. The distance between two candidate entities i and j is computed using *Haversine* formula:

$$d_{ij} = Haversine(\varphi_i, \varphi_j, \lambda_i, \lambda_j, r), \tag{2}$$

where (φ_i, λ_i) are the latitude and longitude of entity i and r is the radius of Earth. The distance is successively scaled in the [-1, 1] interval and embedded in an array of dimension d_{dist}, as follows:

$$Emb(d_{ij}) = \alpha_{dist}^{\top}\left(\frac{2 \cdot d_{ij}}{max\_dist} - 1\right) + \beta_{dist}, \tag{3}$$

where $\alpha_{dist} \in \mathbb{R}^{d_{dist}}$ and $\beta_{dist} \in \mathbb{R}^{d_{dist}}$ are learnable parameters and max_dist is the maximum distance between candidate entities chosen during the blocking phase.

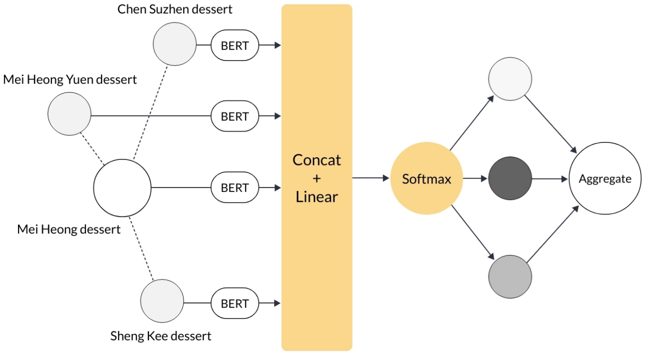


Figure 3: Additive attention mechanism applied to nearby entities: the central entity (white) attends its neighbours (light grey) and computes different attention scores; context vector is created by aggregating the neighbour entities using their attention score

3.5 Neighbourhood Attention

We observe that in some cases, two entities sharing a very similar name and being located a short distance away still cannot be deemed a match due to several factors. First, distance is sometimes affected by the Geo-positional system inaccuracy. Second, a popular restaurant chain, for example, may have many businesses in a small dense area; on the other hand, a small business with a unique name is more likely to match with another with the same name, even if the distance is not very precise. Moreover, for wide spatial entities, like airports or parks, the distance may be very large, but the two entities could still match. The last example in Figure 1a, shows two *non-matching* POIs that have the same name and are very close in space. Using textual comparison and distance alone, the two POIs cannot be classified correctly. Figure 1b shows that the popular chain of coffeehouses *Starbucks* has four distinct businesses in the POI-dense area of Orchard Road. The smaller pins in the picture are neighbours of the candidate entities, retrieved from the two datasets that are being joined. In the example, all the neighbours are *Starbucks* POIs and are spatially close to the candidate entities, delivering useful information about the density of the district and the type (e.g., a chain) of the POI itself.

We argue that the best way to face the challenge described above, is to observe the neighborhood of the entities we are comparing, and take it in account during classification. This approach carries a number of advantages. First, the algorithm can get an overview of the area and figure out if it is a dense district or a sparse suburb. Second, the model can look for entities with a similar name in the same area, that could be a better match: this is a major insight to improve the matching accuracy, since it signals how common a word, or a span of words in the entity name, is in a certain area of the city.

We design a novel Neighbourhood Attention component to embed information from surrounding entities. First, for each candidate entity-pair (e_1, e_2), we obtain a set of neighbours that have similar name and are spatially close, during the *Blocking* phase, using Eq. 1. We remove e_2 from the neighbourhood of e_1 and vice versa, to prevent the candidate entities to pay attention to each other. The names of the central entities as well as the surrounding ones, are summarized using BERT [9]. We use Graph Attention mechanism (GAT) [36] with a single attention head, to obtain a contextualized representation of the surrounding entities. Figure 3 shows our Neighbourhood Attention component. A context vector for the neighbourhood is built comparing the central POI to its neighbours and using additive attention mechanism [2], detailed in the following equations:

$$z_i = W \cdot h_i \tag{4}$$

$$e_{ij} = LeakyReLU(a^{\top}(z_i || z_j)) \tag{5}$$

[2] https://github.com/alvations/sacremoses

Table 1: Number of entities for each city and source

City	#Entities		
	OSM	FSQ	Yelp
Singapore	23,985	31,936	13,699
Edinburgh	11,389	7,549	3,868
Toronto	38,286	18,851	17,204
Pittsburgh	9,387	11,579	6,356

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} exp(e_{ik})} \tag{6}$$

$$n_i = ReLU(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \cdot z_j) \tag{7}$$

Equation 4 is a linear transformation of h_i, which is the BERT-encoding of node i. $W \in \mathbb{R}^{d_{hidden} \times d_{hidden}}$ is the corresponding learnable weight matrix. Equation 5 computes an *unnormalized* attention score between node i and its neighbour j. $||$ denotes concatenation, the attention is parametrized by a weight vector $a \in \mathbb{R}^{2 \cdot d_{hidden}}$ and *LeakyReLU* is the activation function. Equation 6 normalizes the attention scores of the neighbours, using the $Softmax$ function. $\mathcal{N}(i)$ is the set of neighbours of i. Finally, in Equation 7, the neighbours embeddings are aggregated together, weighted by their attention scores and an activation function is applied.

Distance Bias. In Equation 5, the attention scores are computed solely based on the semantic similarity of an entity and its neighbours. In our network of POIs, each node is connected to others with an edge whose weight is given by the distance. In order to include this edge feature into the attention score computation, we decide to add a bias term that depends on the distance, to Eq. 5:

$$e_{ij} = LeakyReLU(a^{\top}(z_i||z_j) + b_{attn}) \tag{8}$$

$$b_{attn} = \phi \cdot \frac{1}{d_{ij}} \tag{9}$$

where d_{ij} is the distance between i and j, and ϕ is a learnable parameter. With this adjustment, the algorithm can learn to pay attention to an entity's neighbours, not only based on the semantic similarity, but also depending on spatial distance.

4 EXPERIMENTS

To evaluate the effectiveness of our proposed method, experiments are conducted on eight real-world datasets from three different sources to compare with existing baselines. To investigate the contribution of each component of our model in the performance gain, we conduct an ablation study. We also perform an analysis of the robustness to show how the algorithm behaves when training data is not available for a specific city.

Table 2: Statistics on the datasets used in the experiments. The column *#Positive* shows the number of positive samples. The column *%Address* shows the ratio of samples where both the entities have non-null address information

Source	City	Size	#Positive (%)	%Address
OSM-FSQ	Singapore	19,243	2,116 (11.0%)	25.5%
	Edinburgh	17,386	3,350 (19.3%)	45.5%
	Toronto	17,858	3,862 (21.6%)	32.0%
	Pittsburgh	5,001	1,454 (29.1%)	25.5%
OSM-Yelp	Singapore	21,588	2,941 (13.6%)	51.6%
	Edinburgh	18,733	2,310 (12.3%)	74.0%
	Toronto	27,969	5,426 (19.4%)	39.9%
	Pittsburgh	5,116	1,622 (31.7%)	38.8%

4.1 Datasets

We collect a total of 194,089 geospatial entities from three real-world location-based services using the respective APIs: Yelp[3], Foursquare (FSQ)[4] and OpenStreetMap (OSM)[5]. For each data source, four cities are considered, namely Singapore, Edinburgh, Toronto, and Pittsburgh. For each entity, we collect a set of textual attributes and the geographical position, as specified in 3.1. The details on the entities collected are shown in Table 1. We asked four human annotators to find and annotate entity matches between OSM and FSQ and between OSM and Yelp. Annotated pairs are used as ground truth in the experiments. Two datasets for each city are created, joining OSM-FSQ and OSM-Yelp. We form the datasets following [25]. We first use the *Blocking* criterion in Eq. 1 to obtain a candidate set C; for each pair in C, if it is present in the set of ground truth matches, we mark it as *match*, else we mark it as a *non-match*. The statistics for each dataset are summarized in Table 2.

4.2 Comparison Methods

We compare our work to state-of-the-art solutions for Entity Resolution and for geospatial data integration. The results are reported in terms of the F1 score on the test set, and the epoch that resulted in the best performance on the validation set.

- **DeepMatcher** [25] is the best performing ER solution that does not rely on pre-trained language models. Its architecture is based on RNNs for attribute values aggregation, attribute comparison, and attention mechanism for attribute soft alignment. It leverages FastText [5] to train the word embeddings.
- **Ditto** [21] is the state-of-the-art solution for entity resolution. It casts ER as a sequence-pair classification and the

[3] https://www.yelp.com/developers
[4] https://developer.foursquare.com/
[5] https://www.openstreetmap.org/

Table 3: Experimental results: bold indicates highest F1 score. Last row shows the improvement of our model with respect to the best baseline

	OSM-FSQ				OSM-Yelp			
	Singapore	Edinburgh	Toronto	Pittsburgh	Singapore	Edinburgh	Toronto	Pittsburgh
DeepMatcher	76.6%	88.2%	78.4%	76.9%	80.2%	92.2%	88.5%	90.5%
SkyEx	72.1%	87.8%	88.0%	85.5%	81.1%	87.6%	91.1%	89.7%
Ditto	82.6%	92.1%	88.2%	88.7%	86.3%	94.4%	91.5%	93.5%
Geo-ER	**89.8%**	**95.7%**	**94.6%**	**92.7%**	**92.9%**	**97.1%**	**96.6%**	**97.6%**
Improvement	+7.2%	+3.6%	+6.4%	+4.0%	+6.6%	2.7%	+5.1%	+4.1%

task is off-loaded to a pre-trained language model. Three optimization techniques are also proposed: summarization, domain knowledge injection and data augmentation. In the experiments, we follow the Ditto paper and fine-tune the RoBERTa [22] language model. We fix the sequence length to its default value (256), turn off summarization, since there are no descriptive attributes for this task, use data augmentation with all the operators applied uniformly at random, and do not inject any domain knowledge for a fair comparison.

- **SkyEx** is a recent effort in geospatial data integration, proposed in the work [18], in which Isaj et al. leverage textual similarity measures, semantic similarity between POIs names using WordNet [23] and spatial similarity to predict if a pair of POIs mentions is a match. The algorithm is based on Pareto optimality and needs no hyperparameters to be tuned.
- **Geo-ER** is the proposed method. We leverage BERT as the language model for textual attributes comparison. We compute and embed distance between the two candidate POIs and use neighbourhood attention to embed information from surrounding POIs.

4.3 Experimental Settings

For each dataset, we randomly split 50%, 20%, and 30% of the samples as the training set, validation set, and testing set, respectively. The split is performed keeping the ratio of positive and negative samples uniformly. We fix the sequence length for the input of the LM to 128 for our model and adjust the format of the input to meet the comparison methods requirements. For DeepMatcher, the attributes are kept in separated columns rather than a text sequence. For SkyEx we generate the attributes using the similarity functions provided by the original paper [18]. In the experiments, the hidden size of BERT is 768, the embedding sizes of distance and neighbourhood embeddings are both set to 256. During the training, the model is optimized by Adam optimizer, with a learning rate of 3e-5, a linearly decreasing learning rate schedule and a batch size of 32. For each experiment, the training phase runs for 10 or 15 epochs (depending on the dataset size) and saves the checkpoint with the best F1 score on the validation set. All the experiments with deep learning frameworks are run on a Nvidia K80 GPU (12GB).

4.4 Performance Analysis

Table 3 shows the experimental results on test data. The algorithm with the highest F1 score on each dataset is highlighted in bold. For clarity, in the last row we show the improvement of Geo-ER over the best performing comparison algorithm. From the results we can see that Geo-ER consistently outperforms the best baseline algorithms by at least 2.7% and up to 7.2%. Based on the results we also make a few comparisons and summarize them as follows.

First, algorithms that leverage pre-trained language models (Geo-ER and Ditto) always deliver better results, regardless of the ability to model spatial information. In fact, DeepMatcher leverages RNNs and soft-alignment mechanisms to compare attributes, but cannot model cross attribute relationships. SkyEx, instead, despite being designed to consider the spatial character of the POIs, still relies on static string similarity measures and compares attributes individually. This result rewards our choice to fine-tune a pre-trained LM, since it demonstrates to be superior to previous approaches on the geospatial integration task. We observe that Geo-ER consistently outperforms Ditto, the best baseline, and this demonstrates the effectiveness of the three components of Geo-ER.

A second interesting insight that emerges from the results is that Geo-ER shows minor improvement with respect to the baselines on datasets with a larger amount of available address information. The last column in Table 2 shows the amount of samples for which both entities contain non-null address information. In both Edinburgh datasets, the address information is more abundant, which would lead to better performance of language modeling approaches, with a difference of only 2.7 and 3.6 points in F1-score. Conversely, in Singapore (OSM-FSQ), Toronto (OSM-FSQ) and Pittsburgh, addresses are more sparse, thus increasing the performance

Table 4: Ablation study: on each experiment, one component is removed and results are reported

	OSM-FSQ				OSM-Yelp				
	Singapore	Edinburgh	Toronto	Pittsburgh	Singapore	Edinburgh	Toronto	Pittsburgh	Avg
Geo-ER	**89.8%**	**95.7%**	**94.6%**	**92.7%**	**92.9%**	**97.1%**	**96.6%**	**97.6%**	-
w/o Neigh. emb.	-1.8%	-1.3%	-2.2%	-0.9%	-1.9%	-1.5%	-1.4%	-2.4%	-1.7%
w/o Dist. emb.	-5.5%	-2.3%	-4.2%	-3.1%	-4.8%	-1.4%	-3.7%	-1.7%	-3.3%

Table 5: Test of robustness: Geo-ER is trained on the dataset indicated on the left and validation and test are performed on the dataset indicated on top. The difference in F1 score is compared to Geo-ER when trained on the same dataset it is tested on

	Singapore	Edinburgh	Toronto	Pittsburgh
Singapore	-	94.9% (-2.2%)	94.7% (-1.9%)	97.0% (-0.6%)
Edinburgh	88.5% (-4.4%)	-	93.9% (-2.7%)	95.6% (-2.0%)
Toronto	90.8% (-2.1%)	96.0% (-1.1%)	-	96.8% (-0.8%)
Pittsburgh	90.4% (-2.5%)	95.7% (-1.4%)	94.2% (-2.4%)	-
Avg ΔF1	-3.0%	-1.6%	-2.3%	-1.1%

Table 6: Test of robustness: Ditto is trained on the dataset indicated on the left and validation and test are performed on the dataset indicated on top. The difference in F1 score is compared to Ditto when trained on the same dataset it is tested on

	Singapore	Edinburgh	Toronto	Pittsburgh
Singapore	-	90.0% (-4.4%)	84.7% (-6.8%)	91.4% (-2.1%)
Edinburgh	76.7% (-9.6%)	-	83.8% (-7.7%)	90.4% (-3.1%)
Toronto	80.4% (-5.9%)	91.3% (-3.1%)	-	93.1% (-0.4%)
Pittsburgh	79.7% (-6.6%)	89.3% (-5.1%)	86.4% (-5.1%)	-
Avg ΔF1	-7.4%	-4.2%	-6.5%	-1.9%

gap between Geo-ER and the baselines from 4.0%, up to 7.2%. This result emphasizes the importance of the geospatial components in our model, especially when only partial or incomplete address information is available.

4.5 Ablation Study

We conduct an ablation study to evaluate the contribution of each part of the model: this is done by comparing our original framework with its variants removing each time a different component. Table 4 reports the results of the ablation study. We observe that the model greatly benefits from the geospatial components, which is consistent with the results report for the performance analysis. Specifically, the distance embedding alone leads to an improvement that ranges from 1.2% to up to 5.5%, with an average of 3.3%. As expected, embedding the distance separately from the language model, gives the algorithm a superior numerical reasoning ability. Similarly, the neighbourhood embedding mechanism significantly enhances the performance, of 1.7% on average, with a minimum of 0.9% and a maximum of 2.4%. In summary, the average improvement achieved by the two components together, across the eight datasets, is 5.0%.

From the ablation study, we can conclude that each component is essential and has an important contribution to the overall results.

4.6 Robustness analysis

In real world applications, labeled data may not be available for a specific place: manually labeling samples and training a different model for each city is very expensive and in many cases infeasible. This set of experiments is to evaluate the robustness of our model when making predictions for a city on which the model was not trained. We also compare the results with Ditto [21], which is the best performing baseline algorithm as shown in the previous experiments. We conduct our analysis on the four datasets obtained by joining OSM and Yelp. The datasets sizes and splits are fixed as in the main experiments. In both Tables 5 and 6 the rows indicate the city that is used for training, and the columns indicate the city that is used for validation and test. From the results in Table 5, we observe that our solution experiences only a slight reduction in terms of F1 score when tested on an unseen city. The largest decrease of performance is observed when testing on the Singapore dataset, and training on other cities, with an average ΔF1 of -3.0%. This might be caused by Singapore being the only Asian city and, even being an English-speaking country, is rich of POIs and street names that are not common in its western counterparts. As shown in Table 6, Ditto faces a much higher decrease in performance, from a minimum of 1.9% up to 7.4%. Ditto relies solely on a pre-trained LM, which is highly affected when unseen words (like POI or street names) appear in the test set. On the other hand, our model leverages two geospatial components that display robustness with respect to the training set.

5 CONCLUSION

We present Geo-ER, an ER system to integrate geospatial data. Experimental results show its advantages over existing approaches. The effectiveness of our solution can be attributed to its semantic understanding, attribute-agnostic architecture, numeracy, and spatial capabilities. The ablation study proves the validity of each component. Robustness analysis emphasizes Geo-ER readiness for real-world use cases. Future work directions include the introduction of blocking phase into the learning framework, to build an end-to-end system, as well as a study on how to improve language models, now very sensitive to human typos.

ACKNOWLEDGMENTS

This research is supported, in part, by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI) (No. AN-GC-2020-017), MOE Tier-2 grant MOE2019-T2-2-181, and MOE Tier-1 grant RG114/19 (S). Dezhong Yao is supported by the National Natural Science Foundation of China under Grant No.62072204 and the Fundamental Research Funds for the Central Universities under Grant 2020kfyXJJS019.

REFERENCES

[1] Rifaat Abdalla. 2016. *Geospatial Data Integration.* 105–124. https://doi.org/10.1007/978-3-319-33603-9_6

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *ArXiv* 1409 (09 2014).

[3] Sandrine Balley, Christine Parent, and Stefano Spaccapietra. 2004. Modelling geographic data with multiple representations. *International Journal of Geographical Information Science* 18 (06 2004), 327–352. https://doi.org/10.1080/13658810410001672881

[4] Nils Barlaug and Jon Atle Gulla. 2020. Neural Networks for Entity Matching. *CoRR* abs/2010.11075 (2020). arXiv:2010.11075 https://arxiv.org/abs/2010.11075

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2016. Enriching Word Vectors with Subword Information. *CoRR* abs/1607.04606 (2016). arXiv:1607.04606 http://arxiv.org/abs/1607.04606

[6] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. 2019. End-to-End Entity Resolution for Big Data: A Survey. *CoRR* abs/1905.06397 (2019). arXiv:1905.06397 http://arxiv.org/abs/1905.06397

[7] Nilesh Dalvi, Vibhor Rastogi, Anirban Dasgupta, Anish Das Sarma, and Tamas Sarlos. 2013. Optimal Hashing Schemes for Entity Matching. In *22nd International World Wide Web Conference, WWW '13*. Rio de Janeiro, Brazil, 295–306. http://dl.acm.org/citation.cfm?id=2488415

[8] Hongzhong Deng, Luo Yun, Yi Liu, and Wang Pu. 2019. Point of Interest Matching between Different Geospatial Datasets. *ISPRS International Journal of Geo-Information* 8 (10 2019), 435. https://doi.org/10.3390/ijgi8100435

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[10] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq R. Joty, Mourad Ouzzani, and Nan Tang. 2017. DeepER - Deep Entity Resolution. *CoRR* abs/1710.00597 (2017). arXiv:1710.00597 http://arxiv.org/abs/1710.00597

[11] Ahmed Elmagarmid, Ihab F. Ilyas, Mourad Ouzzani, Jorge-Arnulfo Quiané-Ruiz, Nan Tang, and Si Yin. 2014. NADEEF/ER: Generic and Interactive Entity Resolution. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data* (Snowbird, Utah, USA) *(SIGMOD '14)*. Association for Computing Machinery, New York, NY, USA, 1071–1074. https://doi.org/10.1145/2588555.2594511

[12] Ahmed Elmagarmid, Panos Ipeirotis, and Vassilios Verykios. 2007. Duplicate Record Detection: A Survey. *Knowledge and Data Engineering, IEEE Transactions on* 19 (02 2007), 1 – 16. https://doi.org/10.1109/TKDE.2007.250581

[13] Donatella Firmani, Barna Saha, and Divesh Srivastava. 2016. Online Entity Resolution Using an Oracle. *Proc. VLDB Endow.* 9, 5 (Jan. 2016), 384–395. https://doi.org/10.14778/2876473.2876474

[14] Cheng Fu, Xianpei Han, Jiaming He, and Le Sun. 2020. Hierarchical Matching Network for Heterogeneous Entity Resolution. In *IJCAI*. 3665–3671.

[15] Cheng Fu, Xianpei Han, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. 2019. End-to-End Multi-Perspective Matching for Entity Resolution. 4961–4967. https://doi.org/10.24963/ijcai.2019/689

[16] Chaitanya S. Gokhale, Sanjib Das, AnHai Doan, Jeffrey F. Naughton, Narasimhan Rampalli, Jude W. Shavlik, and Xiaojin Zhu. 2014. Corleone: hands-off crowdsourcing for entity matching. *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data* (2014).

[17] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. *CoRR* abs/2003.11080 (2020). arXiv:2003.11080 https://arxiv.org/abs/2003.11080

[18] Suela Isaj, Torben Bach Pedersen, and Esteban Zimányi. 2019. Multi-Source Spatial Entity Linkage. *CoRR* abs/1911.09016 (2019). arXiv:1911.09016 http://arxiv.org/abs/1911.09016

[19] Roula Karam, Franck Favetta, Rima Kilany, and Robert Laurini. 2010. Integration of Similar Location Based Services Proposed by Several Providers. *Communications in Computer and Information Science* 88, 136–144. https://doi.org/10.1007/978-3-642-14306-9_14

[20] Bing Li, Yukai Miao, Yaoshu Wang, Yifang Sun, and Wei Wang. 2021. Improving the Efficiency and Effectiveness for BERT-based Entity Resolution. In *The 35th AAAI Conference on Artificial Intelligence (AAAI 2021)*.

[21] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment* 14, 1 (Sep 2020), 50–60. https://doi.org/10.14778/3421424.3421431

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[23] George Miller, R. Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1991. Introduction to WordNet: An On-line Lexical Database*. 3 (01 1991). https://doi.org/10.1093/ijl/3.4.235

[24] Anthony Morana, Thomas Morel, Bilal Berjawi, and Fabien Duchateau. 2014. GeoBench: a Geospatial Integration Tool for Building a Spatial Entity Matching Benchmark (Demo. In *International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL'2014)*. Dallas, Texas, United States, 533–536. https://hal.archives-ouvertes.fr/hal-01301125

[25] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *Proceedings of the 2018 International Conference on Management of Data* (Houston, TX, USA) *(SIGMOD '18)*. Association for Computing Machinery, New York, NY, USA, 19–34. https://doi.org/10.1145/3183713.3196926

[26] Hao Nie, Xianpei Han, Ben He, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. 2019. Deep Sequence-to-Sequence Entity Matching for Heterogeneous Entity Resolution. In *CIKM*. 629–638.

[27] Ralph Peeters and Christian Bizer. 2021. Dual-Objective Fine-Tuning of BERT for Entity Matching. *Proc. VLDB Endow.* 14 (2021), 1913–1921.

[28] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[29] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

[30] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

[31] Michael Schäfers and Udo W. Lipeck. 2014. SimMatching: Adaptable Road Network Matching for Efficient and Scalable Spatial Data Integration. In *Proceedings of the 1st ACM SIGSPATIAL PhD Workshop* (Dallas/Fort Worth, Texas) *(SIGSPATIAL PhD '14)*. Association for Computing Machinery, New York, NY, USA, Article 5, 5 pages. https://doi.org/10.1145/2694859.2694866

[32] Vivek R. Shivaprabhu, Booma Sowkarthiga Balasubramani, and Isabel F. Cruz. 2017. Ontology-Based Instance Matching for Geospatial Urban Data Integration.

In *Proceedings of the 3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics* (Redondo Beach, CA, USA) *(UrbanGIS'17)*. Association for Computing Machinery, New York, NY, USA, Article 8, 8 pages. https://doi.org/10.1145/3152178.3152186

[33] Paulo Tabarro, Jacynthe Pouliot, Richard Fortier, and Louis-Martin Losier. 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* XLII-4/W7 (10 2017), 43–48. https://doi.org/10.5194/isprs-archives-XLII-4-W7-43-2017

[34] Antonio Torralba, Kevin Murphy, W.T. Freeman, and Mark Rubin. 2003. Context-Based Vision System for Place and Object Recognition. *Proceedings of the IEEE International Conference on Computer Vision* 1, 273–280 vol.1. https://doi.org/10.1109/ICCV.2003.1238354

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR* abs/1706.03762 (2017). arXiv:1706.03762 http://arxiv.org/abs/1706.03762

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. arXiv:1710.10903 [stat.ML]

[37] Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. *CoRR* abs/1909.07940 (2019). arXiv:1909.07940 http://arxiv.org/abs/1909.07940

[38] Jiannan Wang, Tim Kraska, Michael J. Franklin, and Jianhua Feng. 2012. CrowdER: Crowdsourcing Entity Resolution. *Proc. VLDB Endow.* 5, 11 (July 2012), 1483–1494. https://doi.org/10.14778/2350229.2350263

[39] Jiannan Wang, Guoliang Li, Jeffrey Xu Yu, and Jianhua Feng. 2011. Entity Matching: How Similar is Similar. *Proc. VLDB Endow.* 4, 10 (July 2011), 622–633. https://doi.org/10.14778/2021017.2021020

[40] Ying Zhang, Puhai Yang, Chaopeng Li, Gengrui Zhang, Cheng Wang, Hui He, Xiang Hu, and Zhitao Guan. 2018. A Multi-Feature Based Automatic Approach to Geospatial Record Linking. *International Journal on Semantic Web and Information Systems* 14 (10 2018), 73–91. https://doi.org/10.4018/IJSWIS.2018100104

[41] Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2019. Semantics-aware BERT for Language Understanding. *CoRR* abs/1909.02209 (2019). arXiv:1909.02209 http://arxiv.org/abs/1909.02209

A APPENDIX

A.1 Data Sources

More details on the sources of data are provided below:

- **Yelp** is a platform for crowd-sourced reviews of venues. It provides a complete record of name, address, zip code, and coordinates for each POI.
- **Foursquare** is a Location-based Social Network. Users gain points and rewards performing *check-ins* in the venues they attend. The dataset provides anonymized users' check-in data: the schema of attributes we select is the same as Yelp, but address information is provided much less often.
- **OpenStreetMap** is a platform to collaboratively create a free geographic database of the world. Users can add POIs (amenities), represented as geographical points. We select the same schema as the two previous data sources.

A.2 Address information

In table 2 is shown the percentage of samples where both the entity have *non-null* address information. We add the percentage of POIs with non-null address information for each city and source in table 7.

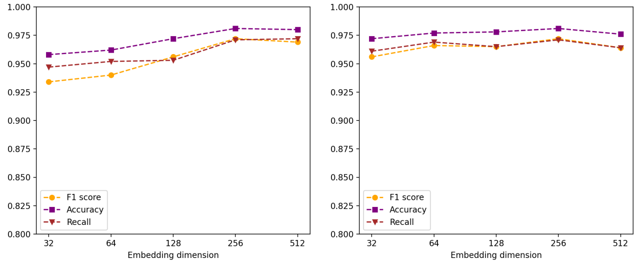
Table 7: Percentage of POIs with *non-null* address information, for each city and source

	% of POIs		
City	OSM	FSQ	Yelp
Singapore	40%	54%	99%
Edinburgh	57%	61%	99%
Toronto	33%	72%	98%
Pittsburgh	28%	58%	97%

A.3 Sensitivity of parameters

We examine the influence of different parameters settings on Geo-ER performance. Specifically, we evaluate how the embedding size of *Distance embedding* and *Neighbourhood embedding* affect Geo-ER in terms of F1-score, accuracy, and recall. We vary both the hyperparameters with values in the set {32, 64, 128, 256, 512}. During the parameters' sensitivity experiments, other parameters remain fixed to their default values. Due to space limits, we only report results relative to the *Pittsburgh OSM-Yelp* dataset. Analogous behavior can be observed on the other datasets.

Figure 4 shows the performance comparison of Geo-ER with different values of Distance Embedding size (4a) and Neighbourhood Embedding size (4b). In general, a larger embedding size grants higher representation capabilities, that lead to better performance. Coherently with the ablation study, showing a higher contribution of the Distance Embedding, we observe that a reduction of its size brings a larger decrease of performance, especially in terms of F1-score. We find 256 to be the best dimension for both the parameters and observe a reduction of the model effectiveness with larger dimensions, probably due to the increasing model complexity and its tendency to overfit.



(a) Distance Embedding **(b) Neighbourhood Embedding**

Figure 4: Sensitivity of parameters