Mining Geospatial Relationships from Text

PASQUALE BALSEBRE, Nanyang Technological University, Singapore
DEZHONG YAO, Huazhong University of Science and Technology, China
GAO CONG, Nanyang Technological University, Singapore
WEIMING HUANG, Nanyang Technological University, Singapore
ZHEN HAI, DAMO Academy, Alibaba Group, Singapore

A geospatial Knowledge Graph (KG) is a heterogeneous information network, capable of representing relationships between spatial entities in a machine-interpretable format, and has tremendous applications in logistics and social networks. Existing efforts to build a geospatial KG, have mainly used sparse spatial relationships, e.g., a district located inside a city, which provide only marginal benefits compared to a traditional database. In spite of the substantial advances in the tasks of link prediction and knowledge graph completion, identifying geospatial relationships remains challenging, particularly due to the fact that spatial entities are represented with single-point geometries, and textual attributes are frequently missing. In this study, we present GTMiner, a novel framework capable of jointly modeling Geospatial and Textual information to construct a knowledge graph, by mining three useful spatial relationships from a geospatial database, in an end-to-end fashion. The system is divided into three components: (1) a Candidate Selection module, to efficiently select a small number of candidate pairs; (2) a Relation Prediction component to predict spatial relationships between the entities; (3) a KG Refinement procedure, to improve both coverage and correctness of a geospatial knowledge graph. We carry out experiments on four cities' geospatial databases, from publicly-available sources and compare with existing algorithms for link prediction and geospatial data integration. Finally, we conduct an ablation study to motivate our design choices and an efficiency analysis to show that the time required by GTMiner for training and inference is comparable, or even shorter, than existing solutions.

CCS Concepts: • **Computing methodologies → Spatial and physical reasoning**; • **Information systems → Graph-based database models**.

Additional Key Words and Phrases: datasets, neural networks, gaze detection, text tagging

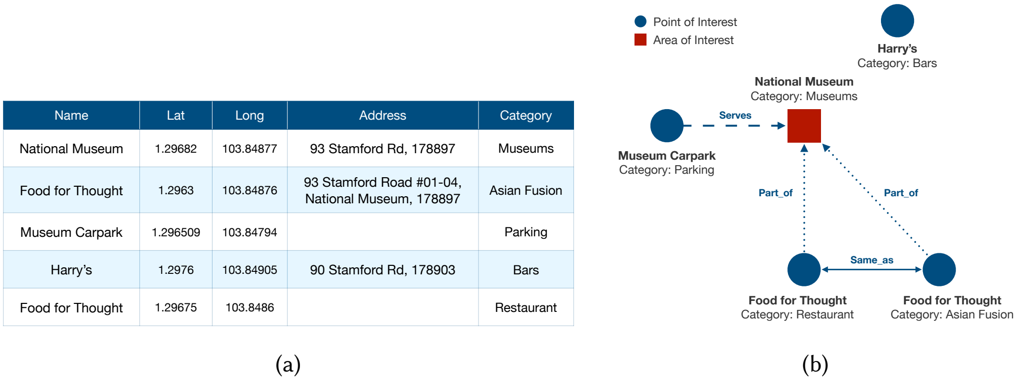
ACM Reference Format:
Pasquale Balsebre, Dezhong Yao, Gao Cong, Weiming Huang, and Zhen Hai. 2023. Mining Geospatial Relationships from Text. *Proc. ACM Manag. Data* 1, 1, Article 93 (May 2023), 26 pages. https://doi.org/10.1145/3588947

1 INTRODUCTION

A Knowledge Graph (KG) is a ubiquitous format of knowledge base, consisting of nodes and directed edges, which stand for the entities and their relations, respectively. The information contained in a KG is usually represented as a set of triples (h, r, t), called facts, where the entity h, the head, is linked to the entity t, the tail, by the relation r. Knowledge graphs have already been successfully applied to several tasks, such as question answering, information retrieval and recommendation systems. In recent years, KGs have been increasingly enriched with geospatial information and efforts have been made to build Geospatial Knowledge Graphs (GKGs) [11, 29]. Existing studies

Authors' addresses: 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; Weiming Huang, Nanyang Technological University, Singapore, weiming.huang@ntu.edu.sg; Zhen Hai, DAMO Academy, Alibaba Group, Singapore, haiz0001@e.ntu.edu.sg.

2836-6573/2023/5-ART93
https://doi.org/10.1145/3588947



Name	Lat	Long	Address	Category
National Museum	1.29682	103.84877	93 Stamford Rd, 178897	Museums
Food for Thought	1.2963	103.84876	93 Stamford Road #01-04, National Museum, 178897	Asian Fusion
Museum Carpark	1.296509	103.84794		Parking
Harry's	1.2976	103.84905	90 Stamford Rd, 178903	Bars
Food for Thought	1.29675	103.8486		Restaurant

(a)

(b)

Fig. 1. (a) Spatial entities contained in a geospatial database. (b) Spatial entities in a Geospatial Knowledge Graph. The entity *Food for Thought* is located inside the *National Museum*. The entity *Museum Carpark*, instead, is located outside the museum but is attached to it. Finally, two records in the database refer to the same entity in the real world.

[20, 36] have applied GKGs to downstream tasks, such as location recommendation, with a discrete success. Still, existing limitations are holding Geospatial Knowledge Graphs back from a wider adoption. In fact, spatial entities, contained in geospatial databases, and used to build GKGs, are described by the combination of textual attributes and geospatial information, which is typically limited to a single point in the space. This representation has proven effective as witnessed by the promising results achieved by many studies in different tasks [14, 23, 68, 74, 75]. Nonetheless, such an approximation hinders the exploration of geospatial relationships.

Figure 1a shows a set of spatial entities contained in a database. In Figure 1b, the same entities, contained in a KG, are linked by the three relationships studied in this paper: *same_as*, *part_of*, and *serves*. The geospatial information, denoted by a latitude and a longitude, can be used to compute an indicative distance between two spatial entities. However, in absence of complex geometries, e.g., polygons, the aforementioned relationships are difficult to identify. Because of this, only sparse spatial relationships have been studied in existing geospatial KGs, such as entities *being inside* cities [29], *closeness* between places [36], and locations *belonging to* classes [11], thus providing little advantage compared to a traditional database. Conversely, the KG depicted in Figure 1b, carries considerable advantages: a user, for instance, can successfully query it to find a museum with a restaurant *inside*, or with a car park *attached*. A search on the database, instead, would be limited to *spatial proximity* between the entities. On the other hand, if a user performs a query for a restaurant, the entity *Food for Thought* may not represent an acceptable result. In fact, being located inside a museum, it may require the user to pay a ticket for the museum, in order to access the restaurant. Similarly, the car park may not be accessible to people outside the museum. In the example in Figure 1, it is clear that the entity *National Museum*, represents an Area of Interest (AOI), affecting its neighboring entities and their functionalities. AOIs, at different granularity levels, have been applied in density-based place clustering [62] and urban region embedding [24, 53]. Consequently, a more articulate representation of the geospatial database is needed, in order to enhance existing systems' capabilities.

In this paper, we introduce the problem of automatic construction of a Geospatial Knowledge Graph, by mining three interesting geospatial relationships from a database, casting it as a Knowledge Graph Completion (KGC) problem. The challenges presented by this task are of different nature.

First, the very limited availability of high-quality polygonal data makes it particularly daunting. Collecting such data is, in fact, vastly expensive due to the requirement of human annotators or land surveying. Furthermore, polygonal data is not a guarantee of success. In Figure 2, the polygonal shape of *Centerpoint Mall*, a shopping center in Toronto, is highlighted in orange. The placeholders indicate the position of Points of Interest from the Yelp database, on the map. All the POIs, in reality, are located inside the shopping center. However, owing to the inaccuracy of the global positioning system, only a subset of them actually falls inside the perimeter of the polygon. This intrinsic limitation of the data requires a solution able to jointly model geospatial information and textual attributes, in order to put the spatial distance in context with respect to the categories and the addresses of the entities. Existing KGC approaches [1, 6, 65] are not designed to process spatial information, and algorithms for geospatial data integration [4, 25] have been proposed to predict only the *equality* relationship between spatial entities. A second challenge is presented by the information contained in a geospatial database, which is often incomplete. Missing columns, and attributes injected under different fields, make it a dirty-data problem. The required information to accurately predict the desired geospatial relations, may be found in a combination of the database fields, therefore the columns cannot be handled independently, and cross-interaction among them is needed. A third challenge is posed by the sparsity of a geospatial KG, whose construction is an *open-world* KGC problem, implying that most of the test entities are not seen during training, consequently an entity's representation should not rely on the known graph topology, but on the entity's textual and geospatial information. The most widely-adopted approaches for KGC [6, 35, 56, 59, 64] are structural-encoding approaches: they assume a *closed-world* setting, producing a meaningful representation for seen entities, in a densely connected graph, and therefore are not suitable for our task. Lastly, records referring to the same real-world entity are common in the geospatial field, especially when data comes from different sources: a system capable of performing resolution of such entities is highly desirable to avoid duplication of results.

To address the aforementioned challenges, we present GTMiner, a system to handle the entire KG construction in an end-to-end fashion. GTMiner first employs a candidate selection step to reduce the number of candidate entities. Subsequently, a KGC algorithm is designed to predict geospatial links between the entities. In order to address the first challenge, we design a Geospatial Encoder to process an entity's spatial information, and a novel Geo-Textual interaction component, such that our KGC model can learn to attend different parts of the textual sequence, depending on the distance between the entities. We adopt a transformer-based Language Model (LM) and exploit the cross-attention mechanism to jointly compare all the entities' textual attributes. Finally, GTMiner has geospatial entity resolution capabilities and is able to recognise, and link, duplicate entities. To further improve the accuracy of the system, we propose a refinement module to recall additional links from previously predicted ones, and delete some that lead to logical inconsistency to achieve higher precision. In summary, this paper makes the following contributions:

- We introduce the task of automatic geospatial Knowledge Graph construction, and propose GTMiner, a system to handle the process in an end-to-end fashion, by mining three geospatial relationships directly from a database.
- We present a novel geospatial Relation Prediction architecture, for open-world KGC, composed of a pre-trained language model, a geospatial encoder and a novel Geo-Textual interaction, to jointly model the textual and geospatial characters of an entity.
- We further propose a KG refinement module, specifically designed to improve both coverage and correctness of a geospatial knowledge graph.
- We introduce four real-world datasets, collected from publicly available sources. We carry out extensive experiments to evaluate the performance of each of our system's components and

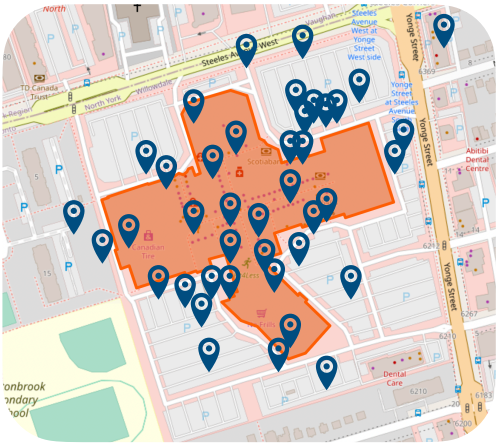


Fig. 2. The polygonal shape of *Centerpoint Mall*, is highlighted in orange. The placeholders are POIs that are located inside the mall. Given the inaccuracy of the global positioning system, only a subset of them actually falls inside the perimeter of the polygon.

perform ablation studies to motivate our design choices. Finally, we share the source code of GTMiner[1], to make our system available for use and our results reproducible.

2 RELATED WORK

2.1 Knowledge Graph Completion

Knowledge Graph Completion is the task of identifying new relationships between the entities in a KG, to form new, plausible triples. KGC methods can be classified into three main approaches: *topology-based*, *textual encoding* and *hybrid*.

2.1.1 Topology-based approach. Topology-based approaches include some of the pioneering studies in the field of KGC. They explore the structure of the graph through measurement in low-dimensional space. TransE [6], a well-known study, is a translation-based approach in which the head entity h is translated in the direction of the relation r, and the distance with the tail t is measured. The $L2$-norm scoring function is given by $-||(\mathbf{h}+\mathbf{r})-\mathbf{t}||$, where bold lower case letters denote vectors. RotatE [54], instead, defines graph embeddings in complex vector space to deal with symmetric relations. TransR [35] learns embeddings for entities and relations in two different vector spaces and TransD [26] decomposes the projection matrix in the product of two vectors, as an improvement over TransR. A different approach is to operate on the whole triple to compute its plausibility score. A well-known effort in this direction is RESCAL [40], whose matching function is $f(h,r,t) = f_r(h,t) = \mathbf{h}^\top \mathbf{W}_r \mathbf{t}$, where $\mathbf{W}_r$ is a relation-specific projection matrix. To reduce the number of parameters, DistMult [64] defines the relation matrix as a diagonal matrix, which in turn makes the approach suitable only to model symmetric relations. ComplEx [56] modifies DistMult to handle also anti-symmetric relations, using complex numbers. SimplE [30] proposes a simple enhancement to tensor factorization to learn *dependently* the two embeddings that represent each entity, when it is the *head* and when it is the *tail* of a triple. ConvE [9] reshapes the embeddings of h, r and t to be 2-dimensional vectors and applies 2D convolutional layers to model the interactions

[1] https://github.com/PasqualeTurin/GTMiner

between entities and relations. Finally, in [28] authors present T-GAP, to model temporal displacement between events in the KG. All the topology-based approaches, learn meaningful embeddings for entities and relations from existing links and from the structure of the known graph. As a consequence, these methods are suitable only for *closed-world* KGC problems, due to the fact that they cannot produce meaningful embeddings for entities (relations) that are not part of the training set, or entities that are not well-connected in the graph. A geospatial KG is a sparse graph, whose construction is an *open-world* KGC problem that involves predicting relations for unseen entities.

2.1.2 Textual encoding approach. Text representation learning is a fundamental problem in natural language processing, which aims at producing a contextualised representation from a sequence of text. Several studies adopted textual encoding for KGC. In particular, they use textual descriptions associated to entities and relations to learn meaningful embeddings, which are subsequently used to predict new links in the graph. Socher et al. [51] use Continuous Bag of Words (CBoW) to represent each component of the triple and a neural tensor network for relation classification. ConMask [50] uses GloVe [43] pre-trained word embeddings to produce a representation for each token in both entities' name and description and for relation's name. It uses attention mechanism to selectively mask entities' description information that is not relevant for the given relation. A CNN is then adopted to produce the latent feature vector that is finally used for classification. Conversely, KG-BERT [65] makes use of a transformer-based pre-trained language model to directly produce a contextual representation of the entire triple, exploiting the cross-attention system embedded in the transformer layers to select pertinent information. Two different versions of the model are presented in the study, namely a and b: the first one receives as input the text of the entire triple (h, r, t) and predicts its plausibility score as a whole; the second receives only the text of the entities and predicts which of the $|\mathcal{R}|$ relations is the most suitable. PKGC [38] builds on KG-BERT(a) and proposes to use a natural language description of the relation, to help the model understand its meaning in the triple. Given their ability to produce a contextualized representation using an entity's textual attributes alone, all the methods based on textual encoding represent an attractive solution for open-world KGC problems.

2.1.3 Hybrid approach. Algorithms to learn knowledge graph embeddings by jointly modeling entities' textual description and graph structure, have been increasingly developed. To allow deep interactions among the two sources of information, Toutanova et al. [55] train continuous representations of structural knowledge and textual relations in a joint fashion and show an improvement over the two individual approaches. Specifically, they build on the DistMult topology-based approach, and add a one-hidden-layer convolutional neural network for textual encoding. The scoring function $f(h, r, t)$ outputs the model's confidence in the existence of the triple. An open-world extension of traditional structural learning methods, is presented in [47], where GloVe word embeddings are added to existing structural solutions, to improve their performance. Recently, StAR [58] proposed to use BERT [10] pre-trained language model and a joint training objective to learn textual and structural representations. In particular, they use a siamese-style BERT to produce a unified encoding for the head entity and the relation $f_{BERT}(h, r)$ and one for the tail entity $f_{BERT}(t)$ and employ a classification objective to maximize positive triples' plausibility and a contrastive objective to minimize the structural distance of the encodings of positive triples. To generate negative samples, positive triples are corrupted, randomly sampling a wrong head or a wrong tail.

2.2 Geospatial Knowledge Graphs

Knowledge Graphs have long enjoyed great popularity in both industry and academia, and famous examples like DBpedia [2], YAGO [52] and Microsoft's Satori[2] have reached a tremendous scale. YAGO is one of the first research projects to construct a knowledge graph automatically, by harvesting entities' information from Wikipedia and combining it with the ontological backbone available in WordNet [39]. The ever-increasing availability of geospatial data, already widely used in applications like recommendation, human mobility and logistics, to name a few, has motivated researchers to adopt KGs in the geospatial field. Radon [49] is a recent effort to explore topological relations, such as *equality*, *intersection* and *enclosure*. It introduces a novel indexing method coupled with a space tiling technique to effectively compute similarity and overlapping of entities' geometries. In contrast, JedAI-spatial [41] after estimating the intersection matrix for a pair of geometries, simultaneously computes all positive topological relations, achieving higher efficiency. Both these approaches operate on polygonal data and assume exact geometries of all the involved spatial entities, i.e. AOIs and POIs, to be known. This assumption substantially reduces the appeal of both algorithms, given the scarcity of high-quality polygonal data, in publicly-available geospatial databases. Geo-ER [4] is a recent approach for geospatial data integration, based on both textual and geospatial encoding. It further proposes a neighbourhood embedding technique to represent information from nearby entities. WorldKG [11] is a comprehensive geographic KG, built from the OpenStreetMap (OSM) dataset. It converts the flat OSM schema of categories into a hierarchical ontology structure, linking spatial objects' categories to the corresponding classes in Wikidata and DBpedia ontologies. YAGO2geo [29] is a new version of YAGO2, with more precise geospatial information. The original KG is enriched by substituting 1-dimensional points of some spatial entities, with line and polygon information from official sources such as administrative divisions. The authors show its effectiveness in answering queries for which precise geospatial information is required. Liu at al. [36] build an Urban Knowledge Graph of Points of Interest. The POIs are linked to categories and brands they belong to, and to districts they are located at, thus creating a graph that reflects semantic similarity and spatial closeness among them, leading to an improvement in location recommendation on some datasets. Hu et al. [20] construct a KG where users are head entities and POIs are tail entities. The relations describe the visits of users to POIs, and embed time information of the visit. Both [20, 36] make use of POI attributes and user-POI interactions, already available in a structured database, e.g., category and neighborhood attributes, to represent *semantic* and *sparse* geospatial relationships. In absence of precise geometries, existing algorithms cannot identify geospatial relationships that require joint spatial and textual reasoning, like the ones studied in this paper; except for Geo-ER, which was specifically designed to accurately identify *same_as* relationships.

2.3 Pre-trained Language Models

Language Models based on the Transformer [57] architecture have established a new state-of-the-art in a variety of NLP tasks [21, 72]. BERT [10], RoBERTa [37] and DistilBERT [46], a smaller and faster version of BERT, are some of the most prominent examples. 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 assume in different contexts. Another advantage is given by the availability of large models, pre-trained on massive text corpora and ready to be fine-tuned on specific tasks.

[2] http://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing

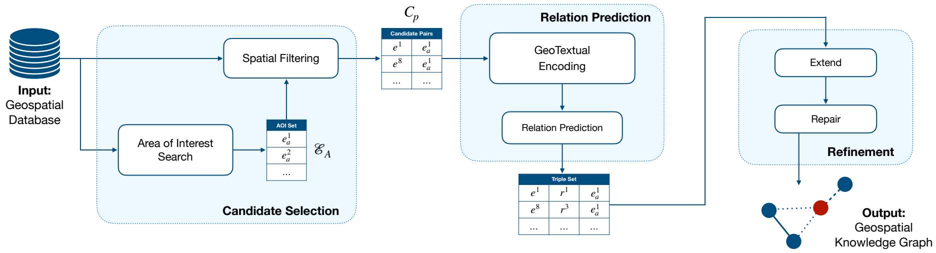


Fig. 3. An overview of the GTMiner framework. It is composed of three main components: (1) the Candidate Selection step (Sec. 4.2), to efficiently select entities likely to share a relation; (2) the Relation Prediction step (Sec. 4.3), to predict geospatial relationships between the entities; and (3) a geospatial KG Refinement algorithm (Sec. 4.4).

3 PROBLEM FORMULATION

A Geospatial Knowledge Graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$ consists of a set of spatial entities with point geometry $\mathcal{E}$, a set of geospatial relationships $\mathcal{R}$, and a set of triples $\mathcal{T} = \{(h, r, t)\}$, where h, t $\in \mathcal{E}$, are linked by $r \in \mathcal{R}$. A spatial entity $e \in \mathcal{E}$ is described by a set of m textual attributes $\{(attr_i : value_i)\}_{1 \leq i \leq m}$, and a location $l_e = (\phi_e, \lambda_e)$, where ϕ_e and λ_e denote the latitude and the longitude of e, respectively. The problem of automatic construction of a geospatial KG, studied in this paper, is stated as follows: given an input geospatial database, as a collection of spatial entities $\mathcal{E}$, and a set of geospatial relationships $\mathcal{R}$, we aim at mining a set of triples $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, in order to construct a KG. In this paper, $\mathcal{R} = \{same\_as, part\_of, serves\}$. The system outputs a set of predicted triples $\mathcal{T}'$, which is compared to the ground truth triples set $\mathcal{T}^*$, and F1-score is used to evaluate its success. Since the overall system's performance depends on the performance of its sub-components, we evaluate each of them separately. The system is generalizable with respect to the set of geospatial relationships $\mathcal{R}$ to be mined, which can be extended, by including additional relationships. We choose only relations that present a clear challenge to be identified, compared to, for instance, *close_to_public_transport*, that can be inferred using a distance filter. At the same time, we do not consider other geospatial relations, such as *beside*, *between*, etc., as they cannot be identified in absence of exact geometries. Finally, the input data may derive from a single source or be merged into one.

4 GTMINER

4.1 System Overview

An overview of the framework is presented in Figure 3. GTMiner receives as input a geospatial database and outputs a geospatial KG, as a set of triples. Several challenges arise in the KG construction process. First, a combinatorial explosion is caused by a model that is required to score every possible triple (h, r, t), to identify if a relation r exists between any two entities h and t. To alleviate this problem, we design a Candidate Selection step (Sec. 4.2), to efficiently select a subset of entity pairs, likely to share a geospatial relationship. Subsequently, a KGC algorithm (Sec. 4.3) predicts a relationship for each entity pair. Given the dirty-data setting of our problem, we decide to adopt pre-trained LMs for the textual encoding part (Sec. 4.3.1), in order to fully exploit the attention-based selectivity of the transformer architecture. Moreover, considering the ability of such models to deal with text-heavy data, comparing attributes individually becomes unnecessary. This, makes the framework *schema-agnostic*, which implies that the correct functioning of the system does not require entries to adhere to the same schema. To process geospatial information, we purposely design a Geospatial Encoder (Sec. 4.3.2). In fact, as a result of the inaccuracy of the global positioning system, threshold-based distance rules would not be effective to determine if a geospatial relationship is plausible between two entities. We further add a Geo-Textual interaction component (Sec. 4.3.3), to let the model contextualize the distance with the textual attributes of the entities. Finally, in section 4.4, we describe our geospatial KG refinement algorithms.

4.2 Candidate Selection

A well-known problem in the task of Knowledge Graph Completion, is the combinatorial explosion caused by a trained model, which, given an incomplete triple of the form $(h, r, ?)$, is asked to score all the candidate triples $\{(h, r, t')|t' \in \mathcal{E}\}$, and rank the correct one, (h, r, t^*), as high as possible. This is further exacerbated in textual encoding approaches, due to the increasing adoption of transformer-based language models, whose computational cost grows quadratically with the sequence length. A recent approach [58] proposes to use a siamese-style encoder, keeping the head and the relation together, and isolating the tail, trading off contextualization capabilities of the model, to achieve a higher efficiency. Our experimental results show that separating the head and the tail entities, leads to a significant reduction of performance, because in a dirty-data setting, with several missing fields, the model necessitates to attend different attributes also depending on the ones available in the other entity. Therefore, the correct employment of the attention mechanism among cross-entity attributes is vital to the effectiveness of the model.

The candidate selection step is designed to mitigate the problem of combinatorial explosion, by efficiently selecting a small subset of candidates, to be subsequently processed by a finer-grained model. It is divided into two main components. The first one is to search for Areas of Interest and the second one to select candidate entities that are in spatial proximity. The concept of AOI has long been studied in geospatial information systems [7, 13, 16, 17] and is commonly defined as an area or a region, typically within an urban environment, which attracts people's attention [22, 33]. The geographic entities that are located inside the perimeter of AOIs are often referred as Points of Interest. Moreover, given the large number of people visiting those areas, neighboring entities often serve a purpose related to the AOI, and have been studied in the context of human mobility patterns [5, 67]. To avoid a loss of generality, we define an area of interest as a spatial entity containing at least another entity inside its perimeter. In a geospatial database, AOIs, such as shopping centers or universities, are represented as generic spatial entities, indistinguishable from POIs, therefore we first train a classifier to identify them.

Formally, given a collection of spatial entities $\{e|e \in \mathcal{E}\}$, each characterized by m attribute-value pairs $\{(attr_i : value_i)\}_{1 \leq i \leq m}$, we train a Feature Extractor $\mathcal{F}(e) \in \mathbb{R}^d$, which converts the spatial entity in a d_f-dimensional feature vector representation. The extracted feature vector, is subsequently fed into a binary classifier C to output the final prediction. The Area of Interest Search module takes as input a spatial entity e and outputs $\hat{y} \in [0, 1]$, which is the probability of the entity e being an AOI,

$$\hat{y} = C(\mathcal{F}(e)). \tag{1}$$

As shown in Table 2, there is a strong imbalance between the two classes, i.e., $|\mathcal{E}| \gg |\mathcal{E}_A|$, which would hamper the classifier accuracy, if trained on the entire set. Because of this, we build a dataset $(\mathcal{D}, \mathcal{Y})$, by randomly sampling (without replacement) k negative examples for each positive one. We vary the value of k among $\{1, 3, 5, 7, 10\}$, to search the one that yields the best results on the validation set. Given a training set of entities associated with their ground truth label, parameters of both $\mathcal{F}$ and C can be optimized by applying stochastic gradient descent, and thus trained to detect AOIs, as follows

Algorithm 1 Spatial Filtering

```
Input: $\mathcal{E}$: spatial entities set, $\mathcal{E}_A$: AOIs set from AOI Search module, $f_{dist}$: distance function, $d_{th}$:
    distance threshold, $f_{sim}$: string similarity function, $sim_{th}$: similarity threshold
Output: $C_p$: candidate pair set
$\mathcal{E} \leftarrow \{\mathcal{E}\} - \{\mathcal{E}_A\}$
for each AOI $e_a^k$ in $\mathcal{E}_A$ do
    Initialize empty candidate set for $e_a^k$: $C_{e_a^k}$
    Initialize empty candidate pairs set for $e_a^k$: $C_{p_a^k}$
    for each entity $e^i$ in $\mathcal{E}$ do
        Compute distance $f_{dist}(e_a^k, e^i)$
        if distance $\leq d_{th}$ then
            $C_{e_a^k} \leftarrow C_{e_a^k} + e^i$
            $C_{p_a^k} \leftarrow C_{p_a^k} + (e^i, e_a^k)$
            for each entity $e^j$ in $C_{e_a^k}$ do
                Compute similarity $f_{sim}(e^i, e^j)$
                if similarity $\geq sim_{th}$ then
                    $C_{p_a^k} \leftarrow C_{p_a^k} + (e^i, e^j)$
    $C_p \leftarrow C_p + C_{p_a^k}$
return $C_p$
```

$$\mathcal{F}^*, C^* = \underset{\mathcal{F},C}{\operatorname{argmin}}\, \mathcal{L}_s. \tag{2}$$

$\mathcal{L}_s$ is the loss function to be minimized during the training of the AOI search module. We adopt a weighted binary cross-entropy loss,

$$\mathcal{L}_s = -\frac{1}{N}\sum_{i}^{N} \alpha\, y_i\, log(\hat{y}_i) + \beta\,(1 - y_i)\, log(1 - \hat{y}_i), \tag{3}$$

where α and β are the weights of the positive and the negative classes, respectively. In particular, we choose $\alpha > \beta$, in order to increase the penalization of the algorithm in case of false negatives. In fact, while false positives would only cause an increase of the number of candidates to be handled by the finer-grained model, false negatives would result in entities linked by geospatial relations, not being included in the candidate set, thus hurting the performance. We choose *name* and *category* as the entity attributes used for classification. For the Feature Extractor $\mathcal{F}$ we employ: (1) Recurrent neural networks, specifically Bidirectional Long-Short Term Memory (BiLSTM) [19] units, a solution widely-adopted [48, 66] in NLP to encode variable-length text into features, capable of retaining the words order in the sequence embedding; (2) BERT [10] pre-trained language model. We adopt GloVe [43] pre-trained word embeddings to transform words into vectors, before feeding them to the BiLSTM network. We use a multi-layer perceptron (MLP), as the binary classifier C.

The second component of the candidate selection module aims at retrieving spatial entities in close proximity of Areas of Interest. In fact, geospatial relationships are predominantly observed between entities at a close spatial distance, therefore the efficiency of the system can be highly increased by selecting only neighboring entities as potential candidates. The Spatial Filtering algorithm is illustrated in Algorithm 1. It receives as input the list of entities $\mathcal{E}_A$ classified as AOIs by the previous component, and the complete set of spatial entities $\mathcal{E}$. A distance function (e.g., Haversine, Euclidean...) and a similarity function (e.g., Levenshtein, Jaccard...) are specified, along

Table 1. The geospatial relationships covered in this study, each associated with a short description.

Name	Description
part_of	(a, *part_of*, b) $\longrightarrow$ a is located inside the perimeter of b
same_as	(a, *same_as*, b) $\longrightarrow$ a and b are two spatial records that refer to the same spatial entity in the real world
serves	(a, *serves*, b) $\longrightarrow$ a is a spatial entity that provides a service to the entity b, in terms of human mobility (e.g., a taxi stand, a parking lot...), assistance (e.g., an information desk) etc.
unknown	(a, *unknown*, b) $\longrightarrow$ a special relation indicating that none of the above relationships links a and b.

with two thresholds, d_{th} and sim_{th}. The component outputs a set of candidate pairs C_p. For each AOI e_a^k in the list, each neighboring entity e^i, whose distance is lower than the given threshold d_{th}, is selected (lines 2-9), and added to e_a^k's candidate set $C_{e_a^k}$ (line 8), and the candidate pair (e^i, e_a^k) is added to the candidate pair set $C_{p_a^k}$ (line 9). Furthermore, entities in the same candidate set $C_{e_a^k}$ are compared using the similarity function f_{sim} (lines 10-12), applied on the *Name* attribute. Spatial entities with a name similarity higher than the given threshold sim_{th}, are selected as potential duplicates, and added as candidates in $C_{e_a^k}$ (line 13). Finally, the set of candidate pairs $C_{e_a^k}$, associated with the Area of Interest e_a^k, is appended to the full list of candidate pairs C_p (line 18), which is returned as output (line 20).

4.3 Relation Prediction

The Relation Prediction module aims at discovering geospatial relationships between the entities, so as to construct a Knowledge Graph. This, presents a task-specific challenge. In fact, 1-to-n and n-to-1 relationships exist among the entities. A spatial entity, for instance, may be linked by *same_as* relation to two or more entities. On the other hand, many entities may be linked by *part_of* relation to a single entity. In order to address this challenge, we follow [65] to cast the problem as a relation prediction one: the module receives a small set of candidate entity pairs that are in spatial proximity, retrieved during the candidate selection step, and predicts if a relation exists between two entities, and which one.

Formally, given a set of geospatial relationships $\mathcal{R}$, and a set of candidate entity pairs, in the form of incomplete triples $(h, ?, t)$, the goal of the relation prediction module is to score all candidate triples $\{(h, r', t) | r' \in \mathcal{R}\}$ and to rank the oracle triple (h, r^*, t) as high as possible. We include an additional relationship, called *unknown*, in the set $\mathcal{R}$, in order to enable the model to predict that two candidate entities, retrieved during step 1, do not share any of the defined spatial relationships. An example is the entity *Harry's*, in Figure 1b, that is in the neighbourhood of the AOI *National Museum*, but does not share any of the three relationships with it. All the relations considered in this study, along with a short description, are listed in Table 1.

To predict the relationships between the entities, we design a novel architecture, which is depicted in Figure 4. Three main components are used to perform feature extraction from the spatial entities: (1) a textual encoder, (2) a geospatial encoder, (3) a geo-textual interaction mechanism. Finally, an MLP is used as a multi-class classifier, to project the feature vector to an $|\mathcal{R}|$-dimensional space, for relation prediction.

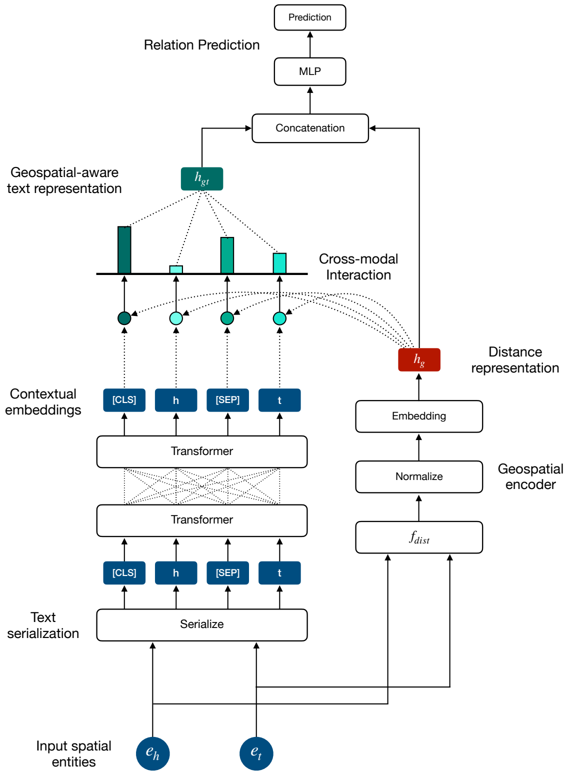


Fig. 4. General architecture of the GTMiner Relation Prediction module. It is composed of a textual encoder (Sec. 4.3.1), a geospatial encoder (Sec. 4.3.2), and a Geo-Textual interaction component (Sec. 4.3.3).

4.3.1 Textual encoder. The textual encoder is the component that processes the textual attributes of the entities. We choose a pre-trained language model as the backbone of this component, given the success of LMs in dealing with text-heavy sequences [65] and in dirty-data settings [4, 34]. Each entity is first serialized as follows:

$$\mathcal{S}_e(e) = \text{[COL]}\ attr_1\ \text{[VAL]}\ value_1 \ldots \text{[COL]}\ attr_m\ \text{[VAL]}\ value_m,$$

where [COL] and [VAL] are the special tokens followed by the column name and the column value, respectively. The two entities' serialized versions are joined by a second serialization step, that operates on the entity pair:

$$\mathcal{S}_p(e_h, e_t) = \text{[CLS]}\ \mathcal{S}_e(e_h)\ \text{[SEP]}\ \mathcal{S}_e(e_t))\ \text{[SEP]},$$

where [CLS] and [SEP] are two special tokens used to classify a textual sequence and to separate entities in the sequence, respectively. The serialized pair is subsequently fed into a pre-trained LM, specifically BERT [10] language model. BERT outputs a tokens' sequence of the same length of input, which is the contextualized representation of the input tokens, obtained with the transformers' cross-attention mechanism applied at token-, attribute- and entity-level. Each token is represented as a d_{hid}-dimensional vector. A common approach in classification problems [34, 58, 65], is to employ a pooling layer, after the LM, that extracts only the first token, corresponding to the [CLS] token, and use it as a representation of the entire sequence for classification. Although successful, this strategy prevents further interaction between the tokens and external signals. Because of this, we pass the full sequence of tokens to the geo-textual interaction component.

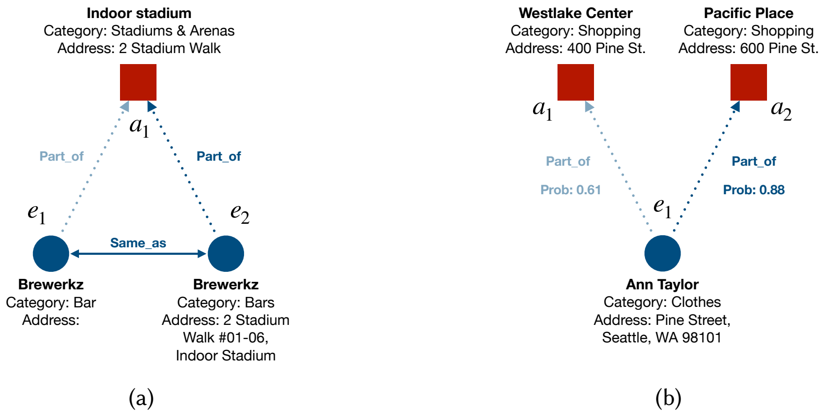
4.3.2 Geospatial encoder. Geospatial information has been successfully integrated in learning-based frameworks by recent studies using different techniques. Zhao et al. [73] divide the map in a fixed-size grid, and learn an embedding for each tile. The distance between two places is then computed as the difference, in the embedding space, between the tiles where the two places are located. POI-Transformers [70] is a recent effort to perform entity matching between Points of Interest's databases. The position of each entity is represented using GeoHash[3] algorithm. It encodes an input position $l = (\phi, \lambda)$ into a fixed-length string. The authors concatenate the GeoHash-generated string to other textual attributes of the POI, and use it as input to a transformer-based language model, to perform deduplication. In contrast, Geo-ER [4] computes the distance between spatial entities using *Haversine* formula, and embeds it into a d-dimensional array, to be concatenated to the output of a language model, achieving higher performance on the same task. We find empirically that embedding directly the distance information leads to better performance, compared to using absolute position. Therefore, we follow Geo-ER, and compute the distance between the head and tail entities, using a distance function f_{dist} (e.g., Haversine, Euclidean...). Then, we normalize its value in the interval $[0, 1]$, and embed it in a d_g-dimensional array. In summary, the output h_g of the geospatial encoder module is

$$\boldsymbol{h}_g = \boldsymbol{\omega}_g \frac{f_{dist}(e_h, e_t)}{d_{th}} + \beta_g, \tag{4}$$

where d_{th} is the distance threshold specified in Algorithm 1, representing the maximum distance between candidate entities. $\boldsymbol{\omega}_g$ and β_g are a vector of learnable parameters and a learnable bias, respectively.

4.3.3 Geo-Textual representation. A widely-adopted approach to train a model on multi-modal data, is to embed each input type in a different high-dimensional space, and subsequently concatenate them [4, 60, 61, 63]. In recent years, cross-modal interaction has emerged as a sophisticated technique that allows multi-modal inputs to jointly share information, leading to significant improvements in fields like NLP [8] and Computer Vision [15, 71]. We design a Geo-Textual interaction component, in order to extend the concept of cross-modal interaction to geospatial data. This allows to contextualize the spatial information with knowledge available in textual form, to correctly predict which relation links two spatial entities. For instance, if a model is to predict if an entity a is located inside an entity b, in absence of cross-modal interaction, the knowledge extracted from the textual attributes would be mostly limited to the address information; the category attribute of the entities would not play a central role in the prediction, while geospatial information would be used to learn a distance threshold. Nonetheless, putting the distance into context with the category of the container entity b (e.g., a shopping mall or an airport), is pivotal to predict if such geospatial relationship exists between the entities. The simple concatenation of textual and geospatial embeddings hinders a similar interaction, thus limiting the performance of the system. The Geo-Textual interaction component is based on the additive attention mechanism [3], to let the model learn to attend different positions in the textual sequence, by jointly attending geospatial and textual information. Formally, given a sequence of N token embeddings $\boldsymbol{T} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, ..., \boldsymbol{t}_N\}$, output of the textual encoder module, and a distance embedding $\boldsymbol{h}_g$, an un-normalized attention score is first computed as:

[3] https://en.wikipedia.org/wiki/Geohash



(a) (b)

Fig. 5. Examples of a missing triple that can be added by our Extension algorithm (a); and a set of two triples causing logical inconsistency, solved by our Repair algorithm (b).

$$\hat{\alpha}_i = \sigma(\boldsymbol{\omega}_a^\top (\boldsymbol{\mathcal{W}}_t^\top \boldsymbol{t}_i || \boldsymbol{\mathcal{W}}_g^\top \boldsymbol{h}_g)), \tag{5}$$

in which, $\boldsymbol{\mathcal{W}}_t \in \mathbb{R}^{d_{hid} \times d_{hid}'}$ and $\boldsymbol{\mathcal{W}}_g \in \mathbb{R}^{d_g \times d_g'}$ are learnable weight matrices, that project the token embedding and the distance embedding, onto the attention space. || denotes concatenation. The attention is parametrized by a learnable weight vector $\boldsymbol{\omega}_a \in \mathbb{R}^{d_{hid}' + d_g'}$, and σ is an activation function. The softmax function is applied to the un-normalized attention scores, over the entire textual sequence length,

$$\alpha_i = Softmax(\hat{\alpha})_i = \frac{e^{\hat{\alpha}_i}}{\sum_{j=1}^{N} \hat{\alpha}_j}. \tag{6}$$

The output vector $\mathbf{h}_{gt} \in \mathbb{R}^{d_{hid}}$ of the Geo-Textual interaction module is computed as a weighted sum of the token embeddings, multiplied by their attention scores:

$$\boldsymbol{h}_{gt} = \sum_{i=1}^{N} \alpha_i \boldsymbol{t}_i \tag{7}$$

Finally, $\boldsymbol{h}_{gt}$ is concatenated to the distance embedding, and a multi layer perceptron is applied to predict the relation that links the two entities.

4.4 Refinement Module

A knowledge graph is unlikely to be fully correct, and a trade-off between coverage and correctness is addressed differently in each KG [69]. To solve this problem, several methods for KG *refinement* have been proposed [12, 42, 45]. In this study, we present a refinement module, composed of two steps, to improve both coverage and correctness of a geospatial knowledge graph. Note that in both steps, we *only* use test relations, that have been predicted during the Relation Prediction phase.

4.4.1 Extend. Knowledge extension, or completion, algorithms [18, 31, 32] aim at increasing the coverage of a knowledge graph. Our extension refinement predicts missing relations between entities, and adds relations that could not have been predicted during the Relation Prediction step. Figure 5a illustrates an example of a triple that is added by the Extend algorithm. The entity e_1's

Table 2. Statistics of the data

City	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{E}_A\rvert$	Triples			Relations			Category	Address
			Train	Valid	Test	*part_of*	*same_as*	*serves*		
Singapore	17092	370	13076	5229	7852	8526	1547	2656	99.79%	67.21%
Toronto	18911	179	8488	3390	5101	5744	1262	1188	99.92%	62.87%
Seattle	10504	500	7906	3162	4747	4257	1138	1215	99.85%	68.06%
Melbourne	13473	190	3058	1220	1839	2675	610	432	99.94%	62.45%

Table 3. AOI Search dataset statistics.

Entities	Singapore	Toronto	Seattle	Melbourne
$\lvert\mathcal{E}_{pos}\rvert$	370	179	500	190
$\lvert\mathcal{E}_{neg}\rvert$	1850	895	2500	950

Table 4. Results of the Area of Interest search task.

	Singapore			Toronto		
Model	P	R	F1	P	R	F1
BiLSTM	89.8	91.16	90.47	82.47	71.55	76.62
BERT	97.56	99.64	**98.59**	96.15	100.0	**98.04**

	Seattle			Melbourne		
Model	P	R	F1	P	R	F1
BiLSTM	91.68	85.93	88.71	84.81	87.2	85.98
BERT	95.1	95.25	**95.17**	90.77	96.41	**93.5**

address information is missing and the triple (e_1, *part_of*, a_1) could not be predicted using entities' positions alone. Nonetheless, e_2 is predicted to be a duplicate of entity e_1, and has a complete address information, including the name of the area of interest a_1 in the address, suggesting that the entity is located inside the stadium. The Extend algorithm uses the triples (e_1, *same_as*, e_2) and (e_2, *part_of*, a_1), to automatically add ($e1$, *part_of*, $a1$). As a result of the added triples, a successful extension refinement is expected to benefit a KGC system in terms of recall.

4.4.2 Repair. KG Repair, or correction, algorithms identify information in the graph that cause logical inconsistency [18, 27]. Examples of information causing logical inconsistency, in a geospatial knowledge graph, are an AOI deemed to be *same_as* a spatial entity, or a spatial entity deemed to be located inside two *non-overlapping* areas of interest. The latter example is depicted in Figure 5b. When two AOIs are in spatial proximity, an entity could be included in both their spatial candidate sets. The entities a_1 and a_2, in Figure 5b, are located in the same road, and share a significant portion of the address information. As a consequence, the entity e_1 is predicted to be *part_of* both a_1 and a_2. The relation prediction module outputs the most likely relation, associated with its probability, and assigns different probabilities to the two triples, due to different distance between the entities. During the Repair step, the triples are analyzed in pairs, in order to find inconsistencies, and, when found, only the triple with highest probability is kept. In the example in Figure 5b, the Repair algorithm deletes the triple (e_1,$part\_of$, a_1), thus improving the precision of the system.

5 EXPERIMENTS

We evaluate the effectiveness of each component of the proposed framework, in the phases of candidate search, geospatial knowledge graph construction and KG refinement. To do so, we use real-world geospatial databases from four cities, and compare our model to the best performing algorithms in the fields of *open-world* KGC and geospatial data integration.

5.1 Datasets

Four real-world datasets are created by collecting 59,080 spatial entities from OpenStreetMap[4] and Yelp[5], using the respective APIs. For both the sources, we collect all the available attributes, specifically *Name*, *Address*, *Latitude*, *Longitude* and *Categories*. The attribute *Zip Code*, when available, is concatenated to the address. For entities belonging to multiple categories, all the categories are included, separated by semicolon. In OSM, each category is represented as a *key-value* pair. For instance, a bus stop has an attribute *highway* with value *bus_stop*. We select only the value (*bus_stop*), as the category of the entity. The statistics of the datasets are presented in Table 2. Under the *Category* column, we report the percentage of entities with at least one category available. Under the *Address* column, instead, we report the percentage of entities with non-null address information. The remaining attributes (*Name* and positional information) are always available. We hired human annotators, to identify and annotate geospatial relationships between the entities, thus forming four sets of triples, one for each city. The first step in the annotation process is the identification of candidate Areas of Interest, using the name, category and shape on the map. Subsequently a spatial filter is used to automatically suggest, to the annotator, entities in proximity of the candidate AOI, from both data sources. Annotators' decisions for all the relationships are based on updated GIS maps and search engines. An entity is annotated to *serve* an AOI if it is physically adjacent to it, and has a clear reference to the AOI in the name or address information. For instance, the bus stop named *NUH*, located outside the AOI *National University Hospital*, is deemed to serve it. Another example is that of the shopping center's parking lots in Figure 2.

5.2 Candidate Selection

In this section we evaluate the performance of the area of interest search module. As discussed in Section 4.2, we implement two design choices for the feature extractor $\mathcal{F}$: a BiLSTM-based recurrent neural network and a pre-trained language model. The classifier C is a multi layer perceptron. For the BiLSTM model, we use 100-dimensional pre-trained GloVe embeddings, we set hidden layer dimension at 64 for each direction, and a dropout rate of 0.1. The max sequence length is set to 16, and out-of-vocabulary words are represented using the average of all GloVe embeddings. For the language model, we use BERT pre-trained LM and set a dropout rate of 0.1. Due to the BERT word-splitting scheme[6], which splits out-of-vocabulary words into sub-words, leading to an increase of the number of tokens, we set a higher max sequence length of 32. We use the output [CLS] token as the feature vector fed to the classifier. We use a learning rate of 3e-5 and a batch size of 32, for both models. We empirically set the distance threshold $d_{th} = 1500m$ (Haversine) and the similarity threshold $sim_{th} = 0.6$ (edit distance) using the training set. Both the thresholds are set to maximise the recall, i.e., avoid filtering out pairs of entities linked by a relationship. Finally,

[4] https://www.openstreetmap.org/
[5] https://www.yelp.com/developers
[6] https://github.com/alvations/sacremoses

we choose the best values of hyper-parameters α, β and k, the number of negative samples for each positive one, using the validation set. The statistics of the classification dataset, created with $k = 5$, are displayed in Table 3.

Table 4 shows that BERT model achieves higher performance than the BiLSTM on all the datasets. We measure the results in terms of F1 score, where the positive class is represented by AOIs. On two datasets, BERT achieves a perfect recall of 100.0, while keeping a high precision. As discussed in Section 4.2, a high recall is desirable to retrieve as many of the true AOIs as possible, while a high precision reduces the size of the candidate set, increasing the efficiency of the system.

5.3 KGC Baselines

We compare GTMiner's Relation Prediction module to algorithms for knowledge graph completion and geospatial data integration: we select *textual-encoding* algorithms (ConMask [50], KG-BERT(a, b) [65], PKGC [38]) to show the effect of different techniques to represent the entities' and relations' textual attributes; *structural-encoding* (TransE [6], SimplE [30]) and *hybrid* (ComplEx-OWE [47], StAR [58]) approaches are included among the baselines to support our choice of neglecting structural information to solve an *open-world* problem. Finally we select *geospatial-aware* (KG-BERT (+GH), Geo-ER [4]) methods, to analyze the importance of relative distance, compared to absolute position, and to highlight the effect of our multi-modal interaction. We do not include an algorithm based on distance information alone since, in our preliminary results, we found that the inaccuracy of locations (shown in Figure 2), makes such a model unacceptably weak. We provide a detailed description of our baselines:

- **TransE** [6] is a translation-based approach in which the head entity h is translated in the direction of the relation r, and the distance with the tail t is measured. We modify its training strategy, by corrupting the relation instead of the tail entity. During evaluation, we let it score all the possible triples, by changing the relation instead of the tail entity, and take the most plausible one as the final prediction.
- **SimplE** [30] is a more recent structural-encoding approach, that jointly learns the embeddings that an entity assumes when it is the head or the tail of a triple. We change its training/evaluation strategies, as in TransE.
- **ConMask** [50] is the best performing *textual-encoding* approach that do not rely on pre-trained language models. It develops a relationship-dependent content masking to learn to attend only the textual parts relevant for the given relation.
- **ComplEx-OWE** [47] is a *hybrid* approach, as it combines GloVe word embeddings for textual encoding with a simple averaging function for aggregation, along with ComplEx algorithm for structural representation.
- **KG-BERT (a)** [65] is the first version of KG-BERT, in which both the entities and the relation are included in the textual sequence, and the pre-trained language model is used as a binary classifier to predict if the triple is correct. During the training phase, it requires negative samples, which are generated through relation corruption, i.e. generating negative triples by replacing relation r with a random relation r'.
- **KG-BERT (b)** [65], instead, includes only the head and tail information in the textual sequence, and uses BERT model as a multi-class classifier to predict the relation that links two entities.
- **PKGC** [38] is a recent *textual-encoding* approach that builds on KG-BERT. It proposes to use a soft prompt, which describes the relation in natural language form, and a support prompt, which includes the additional attributes of the entity, to better contextualize the triple.
- **StAR** [58] is a *hybrid* framework. The authors propose a siamese-style encoder, based on BERT, by keeping the head and the relation together, and isolating the tail. The structural representation

Table 5. Results of KGC on geospatial data. Bold denotes the best performance in terms of F1 score. We show the F1 improvements of GTMiner over the best baseline, and the improvement achieved by our refinement algorithms. The symbol * indicates that the improvement over the best baseline is statistically significant based on a two-sided t-test with p-value $< 10^{-6}$.

	Singapore			Toronto			Seattle			Melbourne		
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
TransE [6]	4.71	16.3	7.29 (± 0.41)	5.28	12.82	7.47 (± 0.28)	4.51	11.07	6.4 (± 0.79)	4.25	9.56	5.88 (± 0.55)
SimplE [30]	6.64	19.27	9.87 (± 0.96)	4.98	18.3	7.82 (± 0.71)	7.75	9.84	8.66 (± 1.05)	6.9	13.33	9.09 (± 0.93)
ComplEx-OWE [47]	60.18	44.29	51.02 (± 0.39)	60.5	41.13	48.97 (± 0.88)	40.21	29.77	34.2 (± 0.87)	66.54	41.9	51.42 (± 1.02)
ConMask [50]	63.28	50.1	55.93 (± 0.46)	67.86	41.66	51.58 (± 0.78)	58.81	40.45	47.98 (± 0.32)	72.89	44.44	55.21 (± 0.6)
KG-BERT (a) [65]	85.38	86.2	80.64 (± 1.31)	77.55	75.46	76.33 (± 0.99)	74.81	70.27	72.39 (± 1.44)	78.1	76.46	77.25 (± 1.71)
KG-BERT (b) [65]	85.80	78.11	81.77 (± 0.7)	82.58	77.21	79.78 (± 1.25)	77.61	69.11	73.02 (± 1.09)	76.44	72.24	74.52 (± 1.95)
PKGC [38]	80.55	73.38	76.79 (± 1.09)	84.13	67.87	75.13 (± 0.91)	78.44	62.58	69.61 (± 0.9)	77.7	73.96	75.78 (± 2.26)
StAR [58]	65.15	72.66	68.7 (± 0.72)	76.48	80.1	78.24 (± 1.51)	60.96	58.24	59.56 (± 0.47)	81.92	83.97	82.93 (± 0.86)
KG-BERT (+GH)	82.99	86.66	84.78 (± 1.11)	86.26	78.01	81.92 (± 1.28)	73.8	78.95	76.28 (± 1.67)	84.11	77.28	80.55 (± 1.23)
Geo-ER [4]	88.27	84.7	86.44 (± 0.88)	87.25	81.74	84.4 (± 1.16)	78.58	78.91	78.74 (± 1.25)	82.6	88.21	85.31 (± 1.47)
GTMiner	90.07	88.15	89.1* (± 1.04)	86.91	88.4	87.64* (± 1.49)	80.56	80.95	80.75* (± 1.21)	87.87	87.86	87.87* (± 1.31)
GTMiner (+Ex)	90.17	89.25	89.65 (± 1.13)	87.0	89.29	88.13 (± 1.39)	80.8	82.37	81.57 (± 1.29)	88.1	88.78	88.24 (± 1.22)
GTMiner (+Ex +Re)	91.33	89.25	**90.27** (± 1.09)	88.08	89.23	**88.66** (± 1.33)	81.27	82.37	**81.81** (± 1.28)	88.27	88.69	**88.47** (± 1.2)
Δ_{F1}			+3.82%			+4.26%			+3.07%			+3.16%

is introduced with a contrastive loss function, which minimizes the distance between the entities, in the embedding space.

- **KG-BERT (+GH)** is a new version of KG-BERT(b) created by us to make it *spatially-aware*. Specifically we use GeoHash to encode the position of each entity into a fixed-size string, and include it as an additional attribute. We set the precision p of GeoHash to 8, which is the precision level that best performed in our experiments. At $p = 8$, each GeoHash identifier refers to a unique region on the earth of size $28.2m \times 19m$.
- **Geo-ER** [4] is a recent study in geospatial data integration, based on BERT language model, a distance embedding component, and a neighborhood attention mechanism. It uses a simple concatenation of the different components, with an MLP. Geo-ER was designed to predict equality relationship. We extend it to predict multiple relations, by modifying the last fully-connected layer.
- **GTMiner** is GTMiner Relation Prediction module, based on pre-trained LMs, a geospatial encoder and the Geo-Textual representation, based on cross-modal attention.
- **GTMiner (+Ex)** is GTMiner proposed framework, with the Extend algorithm applied on the output triples from the Relation Prediction module.
- **GTMiner (+Ex +Re)** is GTMiner proposed framework, with the Extend and Repair algorithms applied on the output triples from the Relation Prediction module.

5.4 Experimental Settings

For each dataset, we randomly split 50%, 20%, and 30% of the samples as the training, validation, and test sets, respectively. During the splitting phase, we keep the ratio of triples for each relation, uniform. StAR and KG-BERT (a) require negative sampling during the training. Following their settings, we sample 3 negative samples for each positive one for StAR, and 1 negative sample for each positive for KG-BERT (a). We use triple corruption on the relation to generate negative samples. In all the experiments, we set the max sequence length to be 128 and the learning rate to be 3e-5 with a linearly decreasing learning rate schedule. In all the methods involving a language model, we use BERT-base pre-trained weights, with a 768-dimensional hidden size and turn on fine-tuning. The embedding size of our geospatial encoder module is set to 128. All the models are run for 10 epochs: at each epoch, the models are validated on the validation set, and the checkpoint with highest F1-score on the validation set, is saved and tested on the test set. We repeat the experiments 10 times, and report the average performance, with standard deviation. The F1-score is computed as the harmonic mean between precision and recall, and a true positive is defined as any correctly classified geospatial relationship, except *unknown*. This particular choice stabilizes the results with respect to the distance threshold d_{th} chosen in the Candidate Selection step (Sec. 4.2). In other words, choosing a large d_{th}, would include in the candidate set, a large number of entities that do not share any geospatial relationship and, being located at a high distance, are trivial to classify, artificially increasing the performance.

5.5 Performance Analysis

Table 5 shows the results on the knowledge graph completion task. As expected, the performance of methods based on structural information alone, is extremely poor. We find that most of the relationships correctly predicted by TransE and SimplE, belong to the *part_of* category, and are for nodes that have a *same_as* relation with a node seen in the training set, being part of the same AOI. This is, in fact, a case where structural information can be of help. Textual-encoding algorithms that leverage pre-trained language models achieve much higher performance than those relying on GloVe word embeddings. ConMask performs better than ComplEx-OWE, despite the latter being a hybrid approach. ConMask uses a more effective word aggregation strategy, based on attention mechanism, compared to ComplEx-OWE, which utilizes a simple average. This shows that a better textual encoding technique improves the performance more than adding structural information. Consistently with the results in [65], KG-BERT (b) delivers better results than (a). Besides, it does not require negative sampling, making the algorithm more efficient than (a). StAR achieves lower performance compared to similar LM-based architectures. As discussed in Section 4.2, it is a siamese-style architecture, and combines a textual-encoding loss, with a topology-based contrastive loss. Cross-entity attention proved to be of crucial importance in the geospatial KGC task. *Geospatial-aware* algorithms achieve better results on all the datasets. Specifically, KG-BERT (+GH) is our modified version of KG-BERT (b), that is able to read the spatial position of the entities in textual form, by means of GeoHash algorithm. GeoHash's hyper-parameter, the precision p, is tuned on the validation set. Both Geo-ER and GTMiner, leverage a pre-trained LM for the textual-encoding part, and have a separate component for the geospatial encoding. The cross-modal interaction introduced in our system, leads to significantly higher results on every dataset. Moreover, the Extend (+Ex) algorithm always delivers higher results in terms of Recall. The Repair (+Re) algorithm, instead, improves the precision, and the correctness of the KG.

Finally, we notice an increase of performance, linked to the quality of the data, specifically the availability of *Address* attribute. The higher performance on the Singapore dataset, compared to Toronto and Melbourne, shows that algorithms able to represent textual attributes benefit from address information. Seattle is the only exception to this. It has the highest percentage of entities with non-null address information, but most of the baselines achieve lower performance on it. Upon closer investigation, we find that in Seattle, a POI may have a different address compared to the building where it is located. In addition, many AOIs are located in close proximity in the busiest districts of the city, thus making the prediction of geospatial relationships more challenging.

Table 6. Ablation study to motivate our design choices. *Baseline* refers to GTMiner's Relation Prediction module, without refinement algorithms.

Model	Singapore	Toronto	Seattle	Melbourne
Baseline	**89.1**	87.64	**80.75**	**87.87**
Dot-Product	88.74 (-0.36%)	**87.81** (+0.17%)	80.71 (-0.04%)	87.48 (-0.39%)
Euclidean	88.25 (-0.85%)	87.32 (-0.32%)	80.55 (-0.2%)	87.19 (-0.68%)
No GT	86.85 (-2.25%)	84.59 (-3.05%)	79.03 (-1.72%)	85.71 (-2.16%)
No GT + No GE	81.92 (-7.18%)	80.58 (-7.06%)	73.11 (-7.64%)	78.09 (-9.78%)

Table 7. The impact of different pre-training strategy and language model size, on our geospatial KGC task.

LM	Singapore	Toronto	Seattle	Melbourne
BERT	89.1	87.64	80.75	**87.87**
DistilBERT	87.96	85.4	80.74	84.96
RoBERTa	88.71	84.45	**81.11**	86.92
RoBERTa Large	**89.98**	**88.26**	76.33	79.21

5.6 Ablation Study

5.6.1 Design choices. In order to motivate our design choices, we conduct an ablation study and report the results in Table 6. The *Baseline* model is our original architecture; to correctly analyze the performance of each design choice, we turn off the Extend and Repair algorithms. The first change we made is in the attention mechanism used to compute the Geo-Textual interaction: we use dot product attention [57] in place of additive attention. Although the difference is not substantial, the additive attention performs slightly better on average, likely because there is no semantic similarity between textual data and distance embedding, and additive attention does not require a high dot product for two vectors to have a high attention score. Secondly we use Euclidean distance as the distance function f_{dist} in place of Haversine formula. As expected, a performance decrease is observed: in fact Haversine formula is more accurate in computing the distance between two points that lay on an irregularly shaped ellipsoid, like the Earth. In *No GT*, we remove the Geo-Textual interaction altogether to showcase the impact of our novel cross-modal interaction mechanism. Finally, we remove the Geospatial Encoder (GE), which forces us to remove also the Geo-Textual interaction, as it is not applicable anymore. The performance drop in this last case is catastrophic, as the model loses completely its geospatial awareness.

5.6.2 Impact of Language Model. In this section we evaluate the performance of our algorithm, using different pre-trained language models, to show the effect of different pre-training approaches. As in the previous section, we turn off both Extend and Repair algorithms. Table 7 illustrates the results of this experiment. We notice that especially in bigger datasets, like Singapore and Toronto, larger language models achieve higher performance. RoBERTa large, has a 1024-dimensional hidden size, compared to the 728-dimensional of the other models. It suffers especially in the datasets

Table 8. Relation-specific performance of Geo-ER and GTMiner. Bold denotes best performance, underline denotes the best Base model.

		Geo-ER			GTMiner		
	Relation	Base	(+E)	(+E +R)	Base	(+E)	(+E +R)
Sin	*part_of*	86.18	86.6	87.03	88.72	89.47	**90.35**
	same_as	88.92	88.92	89.47	90.4	90.4	**90.61**
	serves	85.85	85.85	85.85	**89.77**	**89.77**	**89.77**
Tor	*part_of*	84.24	85.26	85.4	88.06	88.79	**89.15**
	same_as	89.17	89.17	89.41	89.9	89.9	**90.14**
	serves	80.78	80.78	80.78	**83.11**	**83.11**	**83.11**
Sea	*part_of*	76.9	77.71	78.0	80.06	81.35	**81.7**
	same_as	84.12	84.12	**84.26**	83.33	83.33	83.39
	serves	79.98	79.98	79.98	**80.81**	**80.81**	**80.81**
Mel	*part_of*	83.9	84.38	84.48	87.03	87.55	**87.8**
	same_as	90.14	90.14	90.14	90.81	90.81	**91.09**
	serves	87.43	87.43	87.43	**88.93**	**88.93**	**88.93**

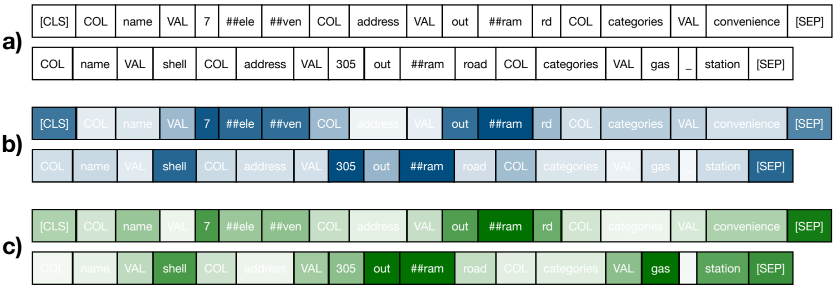
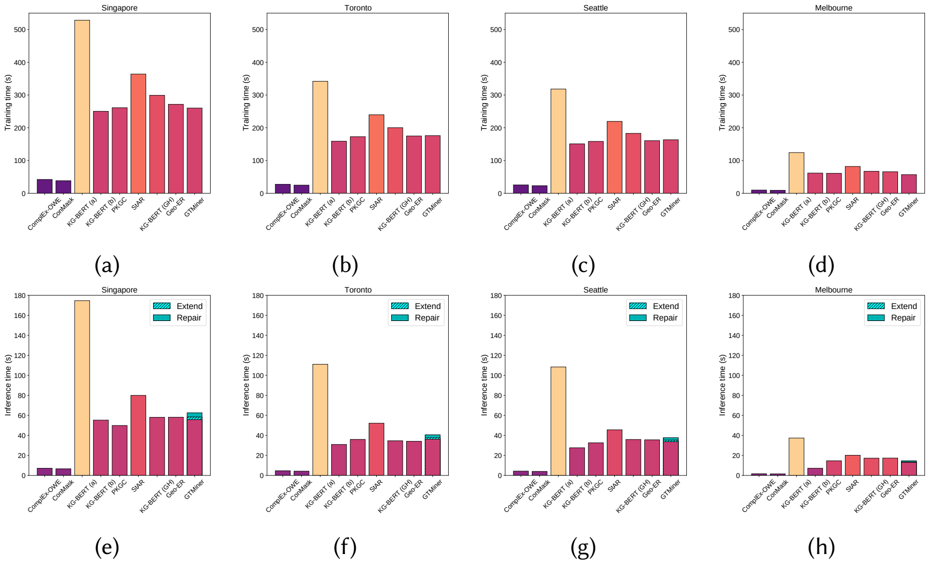

a)
[CLS] COL name VAL 7 ##ele ##ven COL address VAL out ##ram rd COL categories VAL convenience [SEP]
COL name VAL shell COL address VAL 305 out ##ram road COL categories VAL gas _ station [SEP]
b)
[CLS] COL name VAL 7 ##ele ##ven COL address VAL out ##ram rd COL categories VAL convenience [SEP]
COL name VAL shell COL address VAL 305 out ##ram road COL categories VAL gas _ station [SEP]
c)
[CLS] COL name VAL 7 ##ele ##ven COL address VAL out ##ram rd COL categories VAL convenience [SEP]
COL name VAL shell COL address VAL 305 out ##ram road COL categories VAL gas _ station [SEP]

Fig. 6. (a) A triple whose relation is to be predicted. (b) Word relevance scores for Geo-ER. (c) Word relevance scores for GTMiner. '##' is a special identifier used by BERT tokenizer to signal that a word has been split.

smaller in size, with notably lower F1 score. In contrast with existing studies, we decide to use BERT model for all the datasets, in the main experiments, to ensure a fair comparison with the other models.

5.7 Comparison with Geo-ER

Geo-ER is capable of processing textual and geospatial information, and thus represents the closest competitor to GTMiner's Relation Prediction module. Nonetheless, the two information types are simply concatenated, without any form of interaction, limiting the capability of the model to contextualise the distance with information contained in the text. In this section we make a side-by-side comparison to analyze how the two algorithms perform on specific relations, and how the multi-modal interaction improves GTMiner's accuracy. As shown in Table 8, both the algorithms benefit from the refinement module, which rewards GTMiner slightly more, due to the higher number of relationships that are correctly predicted during the Relation Prediction phase. F1-scores are balanced among the classes, and slightly higher on *same_as* relation. While Geo-ER



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7. Efficiency analysis: time required by different models for one training epoch (Fig. a-d); time required by different models for inference on the test set (Fig. e-h).

uses contextual information to improve entity resolution capabilities, outperforming GTMiner on one dataset, the former has an even clearer advantage on the other two spatial relationships.

5.7.1 *Impact of Multi-Modal Interaction.*

Figure 6 shows a triple, whose relation is correctly classified by GTMiner, using the multi-modal interaction, whereas Geo-ER fails. In Figure 6a, we show the plain text of the triple, after tokenization, for readability purposes. The head entity is a *7eleven* convenience store, while the tail entity is a *Shell* gas station. The two entities are located 89m away, and are selected as candidates during the Candidate Selection step. The ground truth is 0 (*unknown*), meaning that none of the 3 spatial relationships holds. Figure 6b shows the importance of each word for Geo-ER decision, using the Lime[7] [44] text explaining library. Although Geo-ER is able to embed distance information, the importance that each word has in the final [*CLS*] embedding is computed independently of the distance. Geo-ER's decision mainly relies on names and addresses, and finally the distance is deemed short enough for the head entity to be located inside the tail, with a final prediction of 2 (*part_of*). In our analysis we notice that Geo-ER often neglects category information, since most instances can be solved without using it. Figure 6c shows the relevance scores, after the multi-modal interaction, that lead GTMiner to output the correct prediction. Even though 89m is a short distance for many AOIs, such as airports and shopping centers, an entity at such distance is unlikely to be located inside a gas station. In accordance with our intuition, the distance has a pivotal role also in the selection of words that will contribute to the final decision.

5.8 Efficiency Analysis

We evaluate the efficiency of the baselines, in terms of training and inference time. Figures 7a-7d illustrate the results. ComplEx-OWE and ConMask are the most efficient ones since they are not based on pre-trained LMs or on LSTM units. On the other hand, KG-BERT (a) and StAR, are the

[7] https://github.com/marcotcr/lime

least efficient since they require negative sampling. Geospatial-aware models are less efficient than KG-BERT (b) and PKGC, but they report much higher performance. KG-BERT (GH) processes longer text sequences, due to the addition of the GeoHash sequence of characters; Geo-ER, instead, uses a neighborhood embedding which involves graph attention; GTMiner is slightly more efficient as it involves a geospatial encoder and a cross-modal attention, whose complexity grows linearly with the sequence length.

Figures 7e-7h illustrate the the time required by each model to perform inference on the test set. The difference between KG-BERT (a) and other models is further exacerbated at prediction time, as it needs to process the candidate entities with each possible relation, and score them by plausibility. We show also the impact of Extend and Repair algorithms on the total inference time. As depicted in the figures, the refinement process weighs only marginally on the test set prediction time.

6 CONCLUSIONS AND FUTURE WORK

In this paper we proposed to mine geospatial relationships from a database of spatial entities. We show that the task is timely and relevant, as a geospatial knowledge graph brings considerable advantages in a great number of applications. We proposed a solution to construct such KG, called GTMiner, composed of three modules, for candidate selection, relation prediction, and KG refinement. Future work directions include building a large-scale, fine-grained, geospatial KG, and designing new algorithms to leverage the geospatial knowledge to offer more accurate recommendations, improve advertising and logistics services.

ACKNOWLEDGMENTS

This research is supported, in part, by MOE Tier-2 grants MOE2019-T2-2-181 and MOE-T2EP20221-0015, and Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI) (No. AN-GC-2020-017). 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. Weiming Huang acknowledges the financial support from the Knut and Alice Wallenberg Foundation.

REFERENCES

[1] Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. 2020. Realistic Re-Evaluation of Knowledge Graph Completion Methods: An Experimental Study. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 1995–2010. https://doi.org/10.1145/3318464.3380599

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*, Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 722–735.

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

[4] Pasquale Balsebre, Dezhong Yao, Gao Cong, and Zhen Hai. 2022. Geospatial Entity Resolution. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 3061–3070. https://doi.org/10.1145/3485447.3512026

[5] Michael Batty. 2007. *Cities and complexity: understanding cities with cellular automata, agent-based models, and fractals.* The MIT press.

[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., Red Hook, NY, USA, 2787–2795.

[7] Debasish Chakraborty. 2015. Web Based GIS Application using Open Source Software for Sharing Geospatial Data (IJARSG, pp 1124-1228). *international journal of advanced remote sensing and gis* 4 (09 2015), 1224–1228. https://doi.org/10.23953/cloud.ijarsg.109

[8] Yixuan Chen, Dongsheng Li, Peng Zhang, Jie Sui, Qin Lv, Lu Tun, and Li Shang. 2022. Cross-Modal Ambiguity Learning for Multimodal Fake News Detection. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 2897–2905. https://doi.org/10.1145/3485447.3511968

[9] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence* (New Orleans, Louisiana, USA) *(AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, Article 221, 8 pages.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[11] Alishiba Dsouza, Nicolas Tempelmeier, Ran Yu, Simon Gottschalk, and Elena Demidova. 2021. *WorldKG: A World-Scale Geographic Knowledge Graph.* Association for Computing Machinery, New York, NY, USA, 4475–4484. https://doi.org/10.1145/3459637.3482023

[12] Islam Akef Ebeid, Majdi Hassan, Tingyi Wanyan, Jack Roper, Abhik Seal, and Ying Ding. 2020. Biomedical Knowledge Graph Refinement and Completion using Graph Representation Learning and Top-K Similarity Measure. *CoRR* abs/2012.10540 (2020). arXiv:2012.10540 https://arxiv.org/abs/2012.10540

[13] Konstantinos Evangelidis, Konstantinos Ntouros, Stathis Makridis, and Constantine Papatheodorou. 2014. Geospatial services in the Cloud. *Computers I& Geosciences* 63 (2014), 116–122. https://doi.org/10.1016/j.cageo.2013.10.007

[14] Shanshan Feng, Lucas Vinh Tran, Gao Cong, Lisi Chen, Jing Li, and Fan Li. 2020. *HME: A Hyperbolic Metric Embedding Approach for Next-POI Recommendation.* Association for Computing Machinery, New York, NY, USA, 1429–1438. https://doi.org/10.1145/3397271.3401049

[15] Peng Gao, Haoxuan You, Zhanpeng Zhang, Xiaogang Wang, and Hongsheng Li. 2019. Multi-Modality Latent Interaction Network for Visual Question Answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).*

[16] Rutuja Gurav, Debraj De, Gautam Malviya Thakur, and Junchuan Fan. 2021. Conflation of Geospatial POI Data and Ground-level Imagery via Link Prediction on Joint Semantic Graph. (11 2021). https://doi.org/10.1145/3486635.3491068

[17] Weiguo Han, Zhengwei Yang, Liping Di, and Richard Mueller. 2012. CropScape: A Web Service Based Application for Exploring and Disseminating US Conterminous Geospatial Cropland Data Products for Decision Support. *Computers and Electronics in Agriculture* 84 (06 2012), 111–123. https://doi.org/10.1016/j.compag.2012.03.005

[18] Yuan He, Jiaoyan Chen, Denvar Antonyrajah, and Ian Horrocks. 2021. BERTMap: A BERT-based Ontology Alignment System. *CoRR* abs/2112.02682 (2021). arXiv:2112.02682 https://arxiv.org/abs/2112.02682

[19] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation* 9 (12 1997), 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735

[20] Bojing Hu, Yaqin Ye, Yingqiang Zhong, Jiao Pan, and Maosheng Hu. 2022. TransMKR: Translation-Based Knowledge Graph Enhanced Multi-Task Point-of-Interest Recommendation. *Neurocomput.* 474, C (feb 2022), 107–114. https://doi.org/10.1016/j.neucom.2021.11.049

[21] 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

[22] Yingjie Hu, Song Gao, Krzysztof Janowicz, Bailang Yu, Wenwen Li, and Sathya Prasad. 2015. Extracting and understanding urban areas of interest using geotagged photos. *Computers, Environment and Urban Systems* 54 (11 2015), 240–254. https://doi.org/10.1016/j.compenvurbsys.2015.09.001

[23] Jizhou Huang, Haifeng Wang, Yibo Sun, Miao Fan, Zhengjie Huang, Chunyuan Yuan, and Yawen Li. 2021. HGAMN: Heterogeneous Graph Attention Matching Network for Multilingual POI Retrieval at Baidu Maps. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery I& Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 3032–3040. https://doi.org/10.1145/3447548.3467059

[24] Bo Hui, Da Yan, Wei-Shinn Ku, and Wenlu Wang. 2020. Predicting Economic Growth by Region Embedding: A Multigraph Convolutional Network Approach. In *Proceedings of the 29th ACM International Conference on Information I& Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 555–564. https://doi.org/10.1145/3340531.3411882

[25] 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

[26] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational

Linguistics, Beijing, China, 687–696. https://doi.org/10.3115/v1/P15-1067

[27] Ernesto Jiménez-Ruiz, C. Meilicke, B. Grau, and I. Horrocks. 2013. Evaluating mapping repair systems with large biomedical ontologies. *CEUR Workshop Proceedings* 1014 (01 2013), 246–257.

[28] Jaehun Jung, Jinhong Jung, and U Kang. 2021. Learning to Walk across Time for Interpretable Temporal Knowledge Graph Completion. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery I& Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 786–795. https://doi.org/10.1145/3447548.3467292

[29] Nikolaos Karalis, Georgios M. Mandilaras, and Manolis Koubarakis. 2019. Extending the YAGO2 Knowledge Graph with Precise Geospatial Knowledge. In *SEMWEB*.

[30] Seyed Mehran Kazemi and David Poole. 2018. SimplE Embedding for Link Prediction in Knowledge Graphs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 4289–4300.

[31] Jiseong Kim, Eun kyung Kim, Yousung Won, Sangha Nam, and Key-Sun Choi. 2015. The Association Rule Mining System for Acquiring Knowledge of DBpedia from Wikipedia Categories. In *NLP-DBPEDIA@ISWC*.

[32] Kristian Kolthoff and Arnab Dutta. 2015. Semantic Relation Composition in Large Scale Knowledge Bases. In *LD4IE@ISWC*.

[33] Chiao-ling Kuo, Ta-Chien Chan, I-Chun Fan, and Alexander Zipf. 2018. Efficient Method for POI/ROI Discovery Using Flickr Geotagged Photos. *ISPRS International Journal of Geo-Information* 7 (03 2018), 121. https://doi.org/10.3390/ijgi7030121

[34] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep Entity Matching with Pre-Trained Language Models. *CoRR* abs/2004.00584 (2020). arXiv:2004.00584 https://arxiv.org/abs/2004.00584

[35] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (Austin, Texas) *(AAAI'15)*. AAAI Press, 2181–2187.

[36] Chang Liu, Chen Gao, Depeng Jin, and Yong Li. 2021. Improving Location Recommendation with Urban Knowledge Graph. *CoRR* abs/2111.01013 (2021). arXiv:2111.01013 https://arxiv.org/abs/2111.01013

[37] 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

[38] Xin Lv, Yankai Lin, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. Do Pre-trained Models Benefit Knowledge Graph Completion? A Reliable Evaluation and a Reasonable Approach. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, Dublin, Ireland, 3570–3581. https://doi.org/10.18653/v1/2022.findings-acl.282

[39] George A. Miller. 1995. WordNet: A Lexical Database for English. *COMMUNICATIONS OF THE ACM* 38 (1995), 39–41.

[40] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning* (Bellevue, Washington, USA) *(ICML'11)*. Omnipress, Madison, WI, USA, 809–816.

[41] Marios Papamichalopoulos, George Papadakis, George Mandilaras, Maria Despoina Siampou, Nikos Mamoulis, and Manolis Koubarakis. 2022. Three-dimensional Geospatial Interlinking with JedAI-spatial.

[42] Heiko Paulheim. 2016. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web* 8 (12 2016), 489–508. https://doi.org/10.3233/SW-160218

[43] 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

[44] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 1135–1144. https://doi.org/10.1145/2939672.2939778

[45] Mohammad Javad Saeedizade, Najmeh Torabian, and Behrouz Minaei-Bidgoli. 2021. KGRefiner: Knowledge Graph Refinement for Improving Accuracy of Translational Link Prediction Methods. *CoRR* abs/2106.14233 (2021). arXiv:2106.14233 https://arxiv.org/abs/2106.14233

[46] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* abs/1910.01108 (2019). arXiv:1910.01108 http://arxiv.org/abs/1910.01108

[47] Haseeb Shah, Johannes Villmow, Adrian Ulges, Ulrich Schwanecke, and Faisal Shafait. 2019. An Open-World Extension to Knowledge Graph Completion Models. *ArXiv* abs/1906.08382 (2019).

[48] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2017. DiSAN: Directional Self-Attention Network for RNN/CNN-free Language Understanding. *CoRR* abs/1709.04696 (2017). arXiv:1709.04696

http://arxiv.org/abs/1709.04696

[49] Mohamed Sherif, Kevin Dreßler, Panayiotis Smeros, and Axel-Cyrille Ngonga Ngomo. 2017. Radon - Rapid Discovery of Topological Relations.

[50] Baoxu Shi and Tim Weninger. 2017. Open-World Knowledge Graph Completion. *CoRR* abs/1711.03438 (2017). arXiv:1711.03438 http://arxiv.org/abs/1711.03438

[51] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with Neural Tensor Networks for Knowledge Base Completion. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., Red Hook, NY, USA, 926–934.

[52] Fabian Suchanek, Gjergji M Kasneci, and Gerhard M Weikum. 2007. Yago: A Core of Semantic KnowledgeUnifying WordNet and Wikipedia. In *16th international conference on World Wide Web (Proceedings of the 16th international conference on World Wide Web)*. Banff, Canada, 697 – 697. https://doi.org/10.1145/1242572.1242667

[53] Ying Sun, Hengshu Zhu, Fuzhen Zhuang, Jingjing Gu, and Qing He. 2018. Exploring the Urban Region-of-Interest through the Analysis of Online Map Search Queries. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery I& Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 2269–2278. https://doi.org/10.1145/3219819.3220009

[54] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *CoRR* abs/1902.10197 (2019). arXiv:1902.10197 http://arxiv.org/abs/1902.10197

[55] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing Text for Joint Embedding of Text and Knowledge Bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1499–1509. https://doi.org/10.18653/v1/D15-1174

[56] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48* (New York, NY, USA) *(ICML'16)*. JMLR.org, 2071–2080.

[57] 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

[58] Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. Structure-Augmented Text Representation Learning for Efficient Knowledge Graph Completion. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 1737–1748. https://doi.org/10.1145/3442381.3450043

[59] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, Québec, Canada) *(AAAI'14)*. AAAI Press, 1112–1119.

[60] Jonatas Wehrmann, Douglas M. Souza, Mauricio A. Lopes, and Rodrigo C. Barros. 2019. Language-Agnostic Visual-Semantic Embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[61] Genta Indra Winata, Zhaojiang Lin, and Pascale Fung. 2019. Learning Multilingual Meta-Embeddings for Code-Switching Named Entity Recognition. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Association for Computational Linguistics, Florence, Italy, 181–186. https://doi.org/10.18653/v1/W19-4320

[62] Dingming Wu, Jieming Shi, and Nikos Mamoulis. 2018. Density-Based Place Clustering Using Geo-Social Network Data. *IEEE Transactions on Knowledge and Data Engineering* 30, 5 (2018), 838–851. https://doi.org/10.1109/TKDE.2017.2782256

[63] Canwen Xu, Feiyang Wang, Jialong Han, and Chenliang Li. 2019. Exploiting Multiple Embeddings for Chinese Named Entity Recognition. *CoRR* abs/1908.10657 (2019). arXiv:1908.10657 http://arxiv.org/abs/1908.10657

[64] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and li Deng. 2014. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. (12 2014).

[65] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for Knowledge Graph Completion. *CoRR* abs/1909.03193 (2019). arXiv:1909.03193 http://arxiv.org/abs/1909.03193

[66] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative Study of CNN and RNN for Natural Language Processing. *CoRR* abs/1702.01923 (2017). arXiv:1702.01923 http://arxiv.org/abs/1702.01923

[67] Yihong Yuan and Raubal Martin. 2012. Extracting Dynamic Urban Mobility Patterns from Mobile Phone Data. 354–367. https://doi.org/10.1007/978-3-642-33024-7_26

[68] Zixuan Yuan, Hao Liu, Junming Liu, Yanchi Liu, Yang Yang, Renjun Hu, and Hui Xiong. 2021. Incremental Spatio-Temporal Graph Learning for Online Query-POI Matching. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 1586–1597. https://doi.org/10.1145/3442381.3449810

[69] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo c, Jens Lehmann, and Sören Auer. 2013. Quality Assessment Methodologies for Linked Open Data. *Semantic Web Journal* (11 2013).

[70] Jinbao Zhang, Changwang Zhang, Xiaojuan Liu, Xia Li, Weilin Liao, Penghua Liu, Yao Yao, and Jihong Zhang. 2022. POI-Transformers: POI Entity Matching through POI Embeddings by Incorporating Semantic and Geographic Information. (2022). https://openreview.net/forum?id=A209HjoI2fq

[71] Zhu Zhang, Zhijie Lin, Zhou Zhao, and Zhenxin Xiao. 2019. Cross-Modal Interaction Networks for Query-Based Moment Retrieval in Videos. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 655–664. https://doi.org/10.1145/3331184.3331235

[72] 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

[73] Ji Zhao, Dan Peng, Chuhan Wu, Huan Chen, Meiyu Yu, Wanji Zheng, Li Ma, Hua Chai, Jieping Ye, and Xiaohu Qie. 2019. Incorporating Semantic Similarity with Geographic Correlation for Query-POI Relevance Learning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence* (Honolulu, Hawaii, USA) *(AAAI'19/IAAI'19/EAAI'19)*. AAAI Press, Article 157, 8 pages. https://doi.org/10.1609/aaai.v33i01.33011270

[74] Kangzhi Zhao, Yong Zhang, Hongzhi Yin, Jin Wang, Kai Zheng, Xiaofang Zhou, and Chunxiao Xing. 2021. Discovering Subsequence Patterns for next POI Recommendation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (Yokohama, Yokohama, Japan) *(IJCAI'20)*. Article 445, 7 pages.

[75] Pengpeng Zhao, Haifeng Zhu, Yanchi Liu, Zhixu Li, Jiajie Xu, and Victor S. Sheng. 2018. Where to Go Next: A Spatio-temporal LSTM model for Next POI Recommendation. *CoRR* abs/1806.06671 (2018). arXiv:1806.06671 http://arxiv.org/abs/1806.06671

Received July 2022; revised October 2022; accepted November 2022