



Iterative Geographic Entity Alignment with Cross-Attention

Alishiba Dsouza^{1,2}, Ran Yu^{1,2}, Moritz Windoffer¹,
and Elena Demidova^{1,2}

¹ Data Science & Intelligent Systems (DSIS), University of Bonn, Bonn, Germany

{dsouza, elena.demidova}@cs.uni-bonn.de, {ran.yu, s5mowind}@uni-bonn.de

² Lamarr Institute for Machine Learning and Artificial Intelligence, Bonn, Germany

<https://lamarr-institute.org/>

Abstract. Aligning schemas and entities of community-created geographic data sources with ontologies and knowledge graphs is a promising research direction for making this data widely accessible and reusable for semantic applications. However, such alignment is challenging due to the substantial differences in entity representations and sparse interlinking across sources, as well as high heterogeneity of schema elements and sparse entity annotations in community-created geographic data. To address these challenges, we propose a novel cross-attention-based iterative alignment approach called IGEA in this paper. IGEA adopts cross-attention to align heterogeneous context representations across geographic data sources and knowledge graphs. Moreover, IGEA employs an iterative approach for schema and entity alignment to overcome annotation and interlinking sparsity. Experiments on real-world datasets from several countries demonstrate that our proposed approach increases entity alignment performance compared to baseline methods by up to 18% points in F1-score. IGEA increases the performance of the entity and tag-to-class alignment by 7 and 8% points in terms of F1-score, respectively, by employing the iterative method.

Keywords: Geographic Knowledge Graph · Iterative Neural Entity Alignment

1 Introduction

Knowledge graphs provide a backbone for emerging semantic applications in the geographic domain, including geographic question answering and point of interest recommendations. However, general-purpose knowledge graphs such as Wikidata [23], DBpedia [14], and YAGO [19] contain only a limited number of popular geographic entities, restricting their usefulness in this context. In contrast, OpenStreetMap (OSM)¹ is a community-created world-scale geographic

¹ <https://www.openstreetmap.org/>.

² OpenStreetMap, OSM and the OpenStreetMap magnifying glass logo are trademarks of the OpenStreetMap Foundation, and are used with their permission. We are not endorsed by or affiliated with the OpenStreetMap Foundation.

data source containing millions of geographic entities. However, the community-driven nature of OSM leads to highly heterogeneous and sparse annotations at both the schema and instance levels, which lack machine-interpretable semantics and limit the accessibility and reusability of OSM data. Knowledge graphs extracted from OSM and dedicated to geographic entities such as LinkedGeoData [1] and WorldKG [7] focus on a selection of well-annotated geographic classes and entities and do not take full advantage of OSM data. Tighter interlinking of geographic data sources with knowledge graphs can open up the rich community-created geographic data sources to various semantic applications.

Interlinking geographic data sources with knowledge graphs is challenging due to the heterogeneity of their schema and entity representations, along with the sparsity of entity annotations and links between sources. Knowledge graphs such as Wikidata adopt ontologies to specify the semantics of entities through classes and properties. Taking the entity Berlin as an example, Table 1a and 1b illustrate its representation in OSM and Wikidata. The property `wdt:P31` (instance of) in Wikidata specifies the entity type. In contrast, OSM annotates geographic entities using key-value pairs called tags, often without clear semantics. The distinction of whether a key-value pair represents an entity type or an attribute is not provided. For instance, in Table 1, the key `capital` in OSM corresponds to a binary value specifying whether the location is the capital of a country. In contrast, the Wikidata property `wdt:P1376` (*capital of*) is an object property linked to an entity of type *country*. Moreover, user-defined key-value pairs in OSM lead to highly heterogeneous and sparse annotations, where many entities do not have comprehensive annotations and many key-value pairs are rarely reused. Finally, sparse and often inaccurate interlinking makes training supervised alignment algorithms difficult. As illustrated in the example, the values, such as the geo-coordinates of the same real-world entity Berlin, differ between sources. Such differences in representation, coupled with the heterogeneity and sparsity of OSM annotations and the lack of links, make schema and entity alignment across sources extremely challenging.

Recently, several approaches have been proposed to interlink knowledge graphs to OSM at the entity and schema level, to lift the OSM data into a semantic representation, and to create geographic knowledge graphs [1, 6, 13, 21]. For example, LinkedGeoData [1] relies on manual schema mappings and provides high-precision entity alignment using labels and geographic distance for a limited number of well-annotated classes. OSM2KG [21] – a linking method for geographic entities, embeds the tags of geographic entities for entity representation and interlinking. The NCA tag-to-class alignment [6] enables accurate matching of frequent tags to classes, but does not support the alignment of rare tags. The recently proposed WorldKG knowledge graph [7] incorporates the information extracted by NCA and OSM2KG, but is currently limited to the well-annotated geographic classes and entities. Overall, whereas several approaches for linking geographic entities and schema elements exist, they are limited to well-annotated classes and entities, they rely on a few properties and do not sufficiently address the representation heterogeneity and annotation sparsity.

Table 1. An excerpt of the Berlin representation in OSM and Wikidata.

| (a) OSM tags. | | (b) Wikidata triples. <code>wd:Q64</code> identifies Berlin. | | |
|---------------|---------------------------|--|---|--------------------------------|
| Key | Value | Subject | Predicate | Object |
| name | Berlin | <code>wd:Q64</code> | <code>rdfs:label (label)</code> | Berlin |
| place | city | <code>wd:Q64</code> | <code>wdt:P31 (instance of)</code> | <code>wd:Q515 (city)</code> |
| population | 3769962 | <code>wd:Q64</code> | <code>wdt:P1082 (population)</code> | 3677472 |
| way | POINT(52.5183 13.4179) | <code>wd:Q64</code> | <code>wdt:P625 (coordinate location)</code> | 52°31'N, 13°23'E |
| capital | yes | <code>wd:Q64</code> | <code>wdt:P1376 (capital of)</code> | <code>wd:Q183 (Germany)</code> |

In this paper, we propose IGEA – a novel iterative geographic entity alignment approach. IGEA relies on a cross-attention mechanism to align heterogeneous context representations across community-created geographic data and knowledge graphs. This model learns the representations of the entities through the tags and properties and reduces the dependency on specific tags and labels. Furthermore, to overcome the annotation and interlinking sparsity problem, IGEA employs an iterative approach for tag-to-class and entity alignment that starts from existing links and enriches the links with alignment results from previous iterations. We evaluate our approach on real-world OSM, Wikidata, and DBpedia datasets. The results demonstrate that, compared to state-of-the-art baselines, the proposed approach can improve the performance of entity alignment by up to 18% points, in terms of F1-score. By employing the iterative method, IGEA increases the performance of the entity and tag-to-class alignment by 7 and 8% points in terms of F1-score, respectively.

In summary, our contributions are as follows:

- We propose IGEA – a novel iterative cross-attention-based approach to interlink geographic entities, bridging the representation differences in community-created geographic data and knowledge graphs.
- To overcome the sparsity of annotations and links, IGEA employs an iterative method for tag-to-class and entity alignment, with integrated candidate blocking mechanisms for efficiency and noise reduction.
- We demonstrate that IGEA substantially outperforms the baselines in F1-score through experiments on several real-world datasets.

2 Problem Statement

In this section, we introduce the relevant concepts and formalize the problem addressed in this paper.

Definition 1 (Knowledge Graph). A knowledge graph $KG = (E, C, P, L, F)$ consists of a set of entities E , a set of classes $C \subset E$, a set of properties P , a set of literals L and a set of relations $F \subseteq E \times P \times (E \cup L)$.

Entities of knowledge graph KG with geo-coordinates L_{geo} are referred to as geographic entities E_{geo} .

Definition 2 (Geographic Entity Alignment). *Given an entity n from a geographic data source G ($n \in G$), and a set of geographic entities E_{geo} from a knowledge graph KG , $E_{geo} \subseteq KG$, determine the entity $e \in E_{geo}$ such that $sameAs(n, e)$ holds.*

In the example in Table 1, as a result of the geographic entity alignment, Berlin from OSM will be linked to Berlin from Wikidata with a *sameAs* link.

Definition 3 (Geographic Class Alignment). *Given a geographic data source G and a knowledge graph KG , find a set of pairs of class elements of both sources, such that elements in each pair (s_i, s_j) , $s_i \in G$ and $s_j \in KG$, describe the same real-world concept.*

In the example illustrated in Table 1, the tag *place=city* from OSM will be linked to the *city* (wd:Q515) class of Wikidata.

In this paper, we address the task of geographic entity alignment through iterative learning of class and entity alignment.

3 The IGEA Approach

In this section, we introduce the proposed IGEA approach. Figure 1 provides an approach overview. In the first step, IGEA conducts geographic class alignment based on known linked entities between OSM and KG with the NCA approach [6]. The resulting tag-to-class alignment is further adopted for blocking in the candidate generation step. Then IGEA applies the cross-attention-based entity alignment module to the candidate set to obtain new links. IGEA repeats this process iteratively with the resulting high-confidence links for several iterations. In the following, we present the proposed IGEA approach in more detail.

3.1 Geographic Class Alignment

We adopt the NCA alignment approach introduced in [6] to conduct tag-to-class alignment. The NCA approach aligns OSM tags with the KG classes. NCA relies on the linked entities from both sources, OSM and a KG, and trains a neural model to learn the representations of the tags and classes. The NCA model creates the shared latent space while classifying the OSM entities into the knowledge graph classes. NCA then probes the resulting classification model to obtain the tag-to-class alignments. NCA selects all matches above a certain threshold value. After applying NCA, we obtain a set of tag-to-class alignments, i.e., (s_i, s_j) , $s_i \in G$, and $s_j \in KG$.

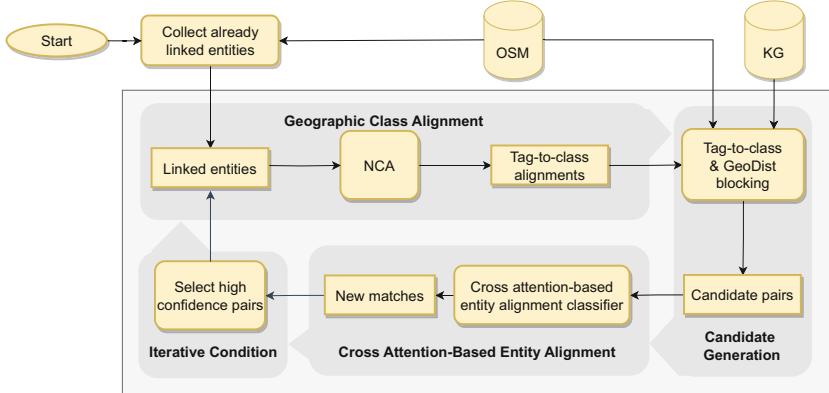


Fig. 1. Overview of the proposed IGEA approach.

3.2 Candidate Generation

OSM contains numerous geographic entities for which we often do not have a match in the KGs. IGEA applies candidate blocking to reduce the search space to make the algorithm more time and complexity efficient. In our task, the objective of the blocking module is to generate a set of candidate entity pairs that potentially match. We built the candidate blocking module based on two strategies, namely entity-type-based and distance-based candidate selection. Entities with a *sameAs* link should belong to the same class. Therefore, we use the tag-to-class alignments produced by the NCA module to select the entities of the same class from both sources to form candidate pairs. Secondly, since we consider only geographic entities, we use spatial distance to reduce the candidate set further and only consider the entities within a threshold distance. Past works observed that a threshold value of around 2000 to 2500 m can work well for most classes [1, 13, 21]. We choose the threshold of 2500 m as mentioned in [21]. The candidate pairs generated after the candidate blocking step are passed to the cross-attention-based entity alignment module.

3.3 Cross-Attention-Based Entity Alignment

We build a cross-attention-based classification model for entity alignment by classifying a pair of entities into a match or a non-match. Figure 2 illustrates the overall architecture of the entity alignment model. The components of the model are described in detail below.

Entity Representation Module: In this module, we prepare entity representations to serve as the model input. For a given OSM node, we select all tags and create a sentence by concatenating the tags. For a given KG entity, we select all predicates and objects of the entity and concatenate all pairs of predicates and objects to form a sentence. We set the maximum length of a sentence to be input to the model to N_w , where N_w is calculated as the average number of

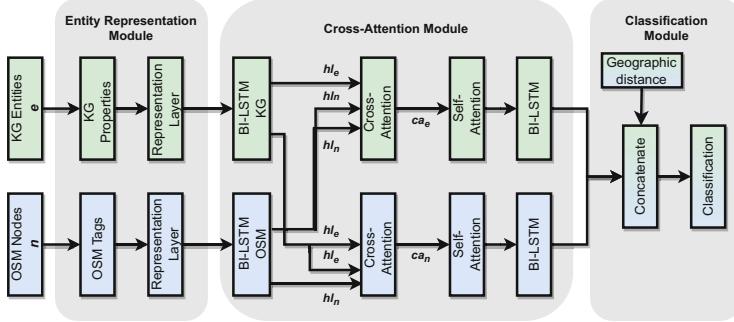


Fig. 2. Cross-attention-based entity alignment model.

words of all entities in the current candidate set. We pass these sentences to the representation layer for each pair of OSM node n and KG entity e .

In the representation layer, the model creates embeddings for the given sentence. We adopt pre-trained fastText word embeddings [3] for the embedding layer. For any word not present in the pre-trained embeddings, we assign a zero vector of size d , where d is the embeddings dimension. In this step, we obtain an array of size $N_w * d$ for each entity.

Cross-Attention Module: We initiate our cross-attention module with a Bi-directional LSTM (BI-LSTM) layer. BI-LSTM models have been demonstrated to perform well on sequential data tasks such as named entity recognition and speech recognition [4, 10]. We adopt BI-LSTM since we want the model to learn to answer what comes after a particular key or a property to help the cross-attention layer. We incorporate BI-LSTM layers after the embedding layers for each of the inputs. As an output, the BI-LSTM layer can return the final hidden state or the full sequence of hidden states for all input words. We select the full sequence of hidden states hl_n, hl_e since we are interested in the sequence and not a single output. These sequences of hidden states hl_n, hl_e are then passed to the cross-attention layer.

Cross-Attention Layer: This layer implements the cross-attention mechanism [22] that helps understand the important properties and tags for aligning the entities. As explained in [22], attention scores are built using keys, values, and queries along with their dimensions. For OSM, we adopt the output of the BI-LSTM layer hl_e as key k and query q and hl_n becomes the value v . For KGs, we adopt the output of the BI-LSTM layer hl_n as key k and query q and hl_e becomes the value v . We initialize the weight vectors w_q, w_k, w_v using the Xavier uniform initializer [9]. We then compute the cross-attention weights for OSM as:

$$Q = hl_e * w_q, K = hl_e * w_k, V = hl_n * w_v,$$

$$att = Q \cdot K, att_w = softmax(att), att_c = att_w \cdot V,$$

where att_w is the attention weights and att_c is the context.

Similarly, we compute the attention weights for KGs by interchanging the values of hl_n and hl_e . We then pass the concatenated att_w and att_c as ca_n and ca_e to the self-attention model.

Self-Attention Layer: Adopting both cross-attention and self-attention layers can improve the performance of the models in multi-modal learning [15]. In our case, the intuition behind adopting the self-attention layer is that the model can learn the important tags and properties of a given entity. The formulation of self-attention is similar to that of cross-attention. Instead of using a combination of outputs from the OSM and KG cross-attention layers ca_n and ca_e , we use only one input, either ca_n and ca_e that is the same across k, q, v . We then pass the self-attention output, i.e., concatenated att_w, att_c , through the final layer of Bi-directional LSTM.

Once we have both inputs parsed through all layers, we concatenate the outputs of the Bi-directional LSTM layers along with the distance input that defines the haversine distance between the input entities.

Classification Module: We utilize the linked entities as the supervision for the classification. Each true pair is labeled one, and the remaining pairs generated by the candidate blocking step are labeled zero. The classification layer predicts whether the given pair is a match or not. We pass the concatenated output through a fully connected layer, which is then passed through another fully connected layer with one neuron to predict the final score. We use a sigmoid activation function with binary cross-entropy loss to generate the score for the final match.

3.4 Iterative Geographic Entity Alignment Approach

We create an end-to-end iterative pipeline for aligning KG and OSM entities and schema elements to alleviate the annotation and interlinking sparsity. We apply the IGEA approach at the country level. For a selected country, we collect all entities having geo-coordinates from the KG. In the first iteration, the already linked entities are used as supervision to link unseen entities that are not yet linked. After selecting candidate pairs and classifying them into match and non-match classes, we use a threshold th_a to only select high confidence pairs from the matched class. In the subsequent iterations, we add these high-confidence matched pairs to the linked entities and then run the pipeline starting from NCA-based class alignment again. By doing so, we aim to enhance the performance of entity alignment with tag-to-class alignment-based candidate blocking and tag-to-class alignment with additional newly linked entities. Algorithm 1 provides details of the IGEA approach.

Algorithm 1 The IGEA Algorithm

Input: n, e OSM and KG linked Entities
 th_a Alignment threshold
 itr number of iterations
 con Country
 kg KG

Output: $align$ Final entity alignment

```

1:  $align \leftarrow \emptyset$ 
2:  $\text{load}(n, e, con)$ 
3:  $KG_e \leftarrow \text{getCountryEntities}(con, kg)$ 
4:  $GT \leftarrow \text{getSeedAlignment}(con, kg)$ 
5: while  $i < itr$  do
6:    $tag-to-class \leftarrow \text{NCA}(con, kg, GT)$ 
7:    $view \leftarrow \text{createView}(tag-to-class)$ 
8:   for all  $ent \in KG_e$  do
9:      $candidates \leftarrow \text{generateCandidates}(ent, view, 2500)$ 
10:    if  $candidates \cap GT \neq \emptyset$  then
11:       $SeenEnt \leftarrow candidates$ 
12:    else
13:       $UnseenEnt \leftarrow candidates$ 
14:    end if
15:   end for
16:    $model \leftarrow \text{classificationModel}(seenEnt)$ 
17:    $prediction \leftarrow model(UnseenEnt)$ 
18:   for all  $pair \in prediction$  do
19:     if  $pair_{confidence} > th_a$  then
20:        $align \leftarrow align \cup \{pair\}$ 
21:        $GT \leftarrow GT \cup \{pair\}$ 
22:     end if
23:   end for
24:    $i = i + 1$ 
25: end while
26: return  $align$ 
```

4 Evaluation Setup

This section describes the experimental setup, including datasets, ground truth generation, baselines, and evaluation metrics. All experiments were conducted on an AMD EPYC 7402 24-Core Processor with 1 TB of memory. We implement the framework in Python 3.8. For data storage, we use the PostgreSQL database (version 15.2). We use TensorFlow 2.12.0 and Keras 2.12.0 for neural model building.

4.1 Datasets

For our experiments, we consider OSM, Wikidata, and DBpedia datasets across various countries, including Germany, France, Italy, USA, India, Netherlands,

Table 2. Ground truth size for Wikidata and DBpedia.

| | France | Germany | India | Italy | Netherlands | Spain | USA |
|----------|--------|---------|-------|-------|-------------|-------|-------|
| WIKIDATA | 19082 | 21165 | 7001 | 16584 | 4427 | 14145 | 73115 |
| DBPEDIA | 10921 | 165 | 1870 | 2621 | 110 | 4319 | 14017 |

and Spain. All datasets were collected in April 2023. For OSM data, we use OSM2pgsql³ to load the nodes of OSM into the PostgreSQL database. The OSM datasets are collected from GeoFabrik download server⁴. For Wikidata⁵ and DBpedia⁶, we rely on the SPARQL endpoints. Given a country, we select all entities that are part of the country with property *P17* for Wikidata and *dbo:country* for DBpedia along with geo-coordinates (*P625* for Wikidata and *geo:geometry* for DBpedia).

4.2 Ground Truth

We select the existing links between geographic entities in OSM and KGs as ground truth. Since we consider geographic entities from the already linked entities identified through “wikidata” and “wikipedia” tags, we select entities with geo-coordinates. Table 2 displays the number of ground truth entities for Wikidata and DBpedia knowledge graphs. We consider only those datasets where the number of links in the ground truth data exceeds 1500 to have sufficient data to train the model. For tag-to-class alignment, we use the same ground truth as in the NCA [6] approach.

4.3 Baselines

This section introduces the baselines to which we compare our work, including similarity-based and deep learning-based approaches.

GeoDistance: In this baseline, we select the OSM node for each KG geographic entity so that the distance between the KG entity and the OSM node is the least compared to all other OSM nodes. We consider the distance calculated using the *st_distance* function of PostgreSQL that calculates the minimum geodesic distance as the distance metric.

LGD [1]: LinkedGeoData approach utilizes geographic and linguistic distance to match the entities in OSM and KG. Given a pair of geographic entities e_1 and e_2 , LinkedGeoData considers $\frac{2}{3}ss(e_1, e_2) + \frac{1}{3}gd(e_1, e_2) > 0.95$ as a match, where ss is the Jaro-Winkler distance and gd is the logistic geographical distance.

³ <https://osm2pgsql.org/>.

⁴ <https://download.geofabrik.de/>.

⁵ <https://query.wikidata.org/>.

⁶ <https://dbpedia.org/sparql>.

Yago2Geo: Yago2Geo [13] considers both string and geographic distance while matching entities by having two filters, one based on Jaro-Winkler similarity (s) between the labels and the second filter based on the Euclidean distance (ed) between the geo-coordinates of the two entities. Given entities $e1$ and $e2$, if $s(e1, e2) > 0.82$ and $ed(e1, e2) < 2000 \text{ meters}$, the two entities are matched.

DeepMatcher: DeepMatcher [17] links two entities from different data sources having similar schema. The model learns the similarity between two entities by summarizing and comparing their attribute embeddings. Since our data sources do not follow the same schema, we select the values of keys name, addressCountry, address, and population for OSM. For KGs, we select the values of the equivalent properties label, country, location, and population.

HierMatcher: This baseline [8] aligns entities by jointly matching at token, attribute, and entity levels. At the token level, the model performs the cross-attribute token alignment. At the attribute level, the attention mechanism is applied to select contextually important information for each attribute. Finally, the results from the attribute level are aggregated and passed through fully connected layers that predict the probability of two entities being a match.

OSM2KG: OSM2KG [21] implements a machine learning-based model for the entity alignment between OSM and KGs. The model generated key-value embeddings using the occurrences of the tags and created a feature vector including entity type and popularity of KG entities. We use the default th_{dist} 2500 m and the random forest classification model adopted in the original paper.

OSM2KG-FT: This baseline is a variation of the OSM2KG model where we replace the key-value embeddings of OSM entities with fastText embeddings.

4.4 Evaluation Metrics

The standard evaluation metrics for entity and tag-to-class alignment are precision, recall, and F1-score computed against a reference alignment (i.e., ground truth). We calculate **precision** as the ratio of all correctly identified pairs to all identified pairs. We calculate **recall** as the fraction of all correctly identified pairs to all pairs in the ground truth alignment. **F1-score** is the harmonic mean of recall and precision. The F1-score is most relevant for our analysis since it considers both precision and recall. We use macro averages for the metrics because we have imbalanced datasets in terms of classes.

5 Evaluation

In this section, we discuss the performance of the IGEA model. First, we evaluate the performance of the approach for entity alignment against baselines.

Furthermore, we assess the impact of the number of iterations and thresholds. Finally, we demonstrate the approach effectiveness on unseen entities through a manual assessment. To facilitate the evaluation, we split our data into 70:10:20 for training, validation, and test data with a random seed of 42.

Table 3. Entity alignment performance on the OSM to Wikidata linking.

| Name | France | | | Germany | | | India | | | Italy | | | Netherlands | | | Spain | | | USA | | |
|-------------|--------|------|-------------|---------|------|-------------|-------|------|-------------|-------|------|-------------|-------------|------|-------------|-------|------|-------------|------|------|-------------|
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| GEODIST | 0.65 | 0.65 | 0.65 | 0.56 | 0.56 | 0.56 | 0.75 | 0.75 | 0.75 | 0.68 | 0.68 | 0.68 | 0.67 | 0.67 | 0.67 | 0.71 | 0.71 | 0.71 | 0.88 | 0.88 | 0.88 |
| LGD | 0.63 | 0.61 | 0.62 | 0.83 | 0.81 | 0.82 | 0.87 | 0.68 | 0.72 | 0.90 | 0.68 | 0.77 | 0.81 | 0.79 | 0.80 | 0.82 | 0.40 | 0.82 | 0.87 | 0.84 | 0.85 |
| YAGO2GEO | 0.5 | 0.51 | 0.50 | 0.53 | 0.51 | 0.50 | 0.61 | 0.60 | 0.60 | 0.52 | 0.51 | 0.50 | 0.50 | 0.88 | 0.64 | 0.63 | 0.70 | 0.65 | 0.88 | 0.69 | 0.73 |
| DEEPMATCHER | 0.62 | 0.58 | 0.60 | 0.74 | 0.67 | 0.71 | 0.77 | 0.79 | 0.78 | 0.89 | 0.55 | 0.68 | 0.83 | 0.78 | 0.80 | 0.87 | 0.75 | 0.80 | 0.93 | 0.91 | 0.91 |
| HIERARMATCH | 0.51 | 0.71 | 0.59 | 0.64 | 0.79 | 0.70 | 0.71 | 0.88 | 0.79 | 0.62 | 0.83 | 0.71 | 0.8 | 0.83 | 0.81 | 0.80 | 0.77 | 0.78 | 0.92 | 0.93 | 0.92 |
| OSM2KG | 0.81 | 0.79 | 0.80 | 0.83 | 0.82 | 0.82 | 0.87 | 0.81 | 0.84 | 0.87 | 0.79 | 0.83 | 0.82 | 0.69 | 0.75 | 0.83 | 0.82 | 0.82 | 0.92 | 0.81 | 0.86 |
| OSM2KG-FT | 0.83 | 0.81 | 0.81 | 0.89 | 0.82 | 0.85 | 0.91 | 0.75 | 0.82 | 0.89 | 0.85 | 0.87 | 0.89 | 0.71 | 0.77 | 0.88 | 0.82 | 0.85 | 0.95 | 0.87 | 0.91 |
| IGEA-1 | 0.95 | 0.91 | 0.94 | 0.93 | 0.95 | 0.94 | 0.88 | 0.87 | 0.87 | 0.93 | 0.97 | 0.94 | 0.94 | 0.86 | 0.90 | 0.89 | 0.91 | 0.90 | 0.93 | 0.95 | 0.94 |
| IGEA-3 | 0.98 | 0.99 | 0.99 | 0.93 | 0.96 | 0.95 | 0.96 | 0.90 | 0.93 | 0.99 | 0.97 | 0.98 | 0.94 | 0.94 | 0.94 | 0.98 | 0.93 | 0.95 | 0.97 | 0.97 | 0.97 |

5.1 Entity Alignment Performance

Tables 3 and 4 present the performance of the IGEA approach and the baselines in terms of precision, recall, and F1-score on the various country datasets for Wikidata and DBpedia knowledge graphs, respectively. IGEA-1 and IGEA-3 indicate the results obtained with the 1st and 3rd iterations of the IGEA approach, respectively. The results demonstrate that the proposed IGEA approach outperforms all the baselines in terms of the F1-score. We achieve up to 18% points F1-score improvement on Wikidata and up to 14% points improvement over DBpedia KGs. IGEA also achieves the best recall and precision on several datasets. Regarding the baselines, as expected, GeoDist performs poorly since the geo-coordinates of the same entity are presented with different precision in OSM and in KGs and are not always in closer proximity to each other. OSM2KG-FT performs the best among the baselines for both KGs. We notice that using the tags with fastText embeddings slightly improves the performance of the OSM2KG over using the occurrence-based key-value embeddings. The deep-learning-based baselines perform on par with the other baselines. The absence of the features such as name and country limits the performance of these deep-learning-based baselines that rely on specific properties. The performance of the name-based baselines such as Yago2Geo and LGD is inconsistent across datasets; a potential reason is the absence of labels in the same language.

Regarding the datasets, the IGEA approach achieved the highest performance improvement on the France and Spain datasets for Wikidata and DBpedia KGs, respectively. The smallest performance improvement over the best-performing baselines is produced on the USA dataset. Data in the USA dataset is mostly in English; furthermore, the USA dataset has the highest percentage of name tags among given countries, which makes string similarity-based baseline

approaches more effective. We notice that India achieves the lowest performance across datasets and KGs. The number of overall properties and tags for entities in India are lower than in other datasets, making IGEA less beneficial. DBpedia results demonstrate better model performance compared to Wikidata. Since DBpedia contains more descriptive properties, it benefits more from employing the cross-attention-based mechanism.

Table 4. Entity alignment performance on the OSM to DBpedia linking.

| Name | France | | | India | | | Italy | | | Spain | | | USA | | |
|-------------|--------|------|-------------|-------|------|-------------|-------|------|-------------|-------|------|-------------|------|------|-------------|
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| GEODIST | 0.39 | 0.39 | 0.39 | 0.35 | 0.35 | 0.35 | 0.58 | 0.58 | 0.58 | 0.40 | 0.40 | 0.40 | 0.64 | 0.64 | 0.64 |
| LGD | 0.84 | 0.76 | 0.79 | 0.83 | 0.63 | 0.72 | 0.87 | 0.69 | 0.76 | 0.91 | 0.72 | 0.78 | 0.70 | 0.61 | 0.64 |
| YAGO2GEO | 0.70 | 0.63 | 0.66 | 0.67 | 0.65 | 0.65 | 0.73 | 0.69 | 0.71 | 0.73 | 0.76 | 0.74 | 0.54 | 0.54 | 0.54 |
| DEEPMATCHER | 0.79 | 0.85 | 0.82 | 0.78 | 0.85 | 0.81 | 0.83 | 0.73 | 0.77 | 0.81 | 0.73 | 0.77 | 0.85 | 0.86 | 0.85 |
| HIERARMATCH | 0.69 | 0.84 | 0.76 | 0.73 | 0.85 | 0.79 | 0.66 | 0.90 | 0.76 | 0.55 | 0.87 | 0.67 | 0.81 | 0.90 | 0.85 |
| OSM2KG | 0.80 | 0.82 | 0.80 | 0.84 | 0.79 | 0.81 | 0.80 | 0.84 | 0.81 | 0.82 | 0.77 | 0.79 | 0.87 | 0.82 | 0.84 |
| OSM2KG-FT | 0.82 | 0.87 | 0.84 | 0.84 | 0.82 | 0.83 | 0.81 | 0.89 | 0.85 | 0.82 | 0.82 | 0.82 | 0.90 | 0.91 | 0.90 |
| IGEA-1 | 0.92 | 0.91 | 0.91 | 0.89 | 0.91 | 0.90 | 0.95 | 0.89 | 0.92 | 0.96 | 0.97 | 0.96 | 0.97 | 0.95 | 0.96 |
| IGEA-3 | 0.95 | 0.99 | 0.97 | 0.96 | 0.97 | 0.97 | 0.95 | 0.98 | 0.96 | 0.96 | 0.95 | 0.95 | 0.99 | 0.97 | 0.98 |

Table 5. Ablation study results for the DBpedia datasets.

| Name | France | | | India | | | Italy | | | Spain | | | USA | | |
|---------------------|--------|------|-------------|-------|------|-------------|-------|------|-------------|-------|------|-------------|------|------|-------------|
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| w/o CROSS-ATTENTION | 0.86 | 0.81 | 0.83 | 0.83 | 0.82 | 0.82 | 0.86 | 0.77 | 0.81 | 0.82 | 0.81 | 0.81 | 0.83 | 0.84 | 0.83 |
| w/o DISTANCE | 0.85 | 0.89 | 0.86 | 0.81 | 0.87 | 0.82 | 0.81 | 0.83 | 0.82 | 0.79 | 0.86 | 0.82 | 0.82 | 0.87 | 0.84 |
| w/o CLASS-BLOCKING | 0.81 | 0.93 | 0.87 | 0.73 | 0.94 | 0.82 | 0.78 | 0.93 | 0.85 | 0.75 | 0.92 | 0.83 | 0.79 | 0.96 | 0.86 |
| IGEA-3 | 0.95 | 0.99 | 0.97 | 0.96 | 0.97 | 0.97 | 0.95 | 0.98 | 0.96 | 0.96 | 0.95 | 0.95 | 0.99 | 0.97 | 0.98 |

5.2 Ablation Study

Table 5 displays the results of an ablation study to better understand the impact of individual components. We observe that removing the cross-attention layer significantly reduces the performance of the model. The class-based blocking improves the recall but has a sharp decrease in precision, as it creates many noisy matches. Removing geographic distance also results in worse performance compared to the IGEA. The results of the ablation study confirm that the components introduced in the IGEA approach help to achieve the best performance.

5.3 Impact of the Number of Iterations

In this section, we evaluate the impact of the number of iterations on the IGEA performance. Figure 3 displays the F1-scores for the entity alignment after each

iteration. We observe that the scores increase in all configurations with the increased number of iterations; after the 3rd iteration, the trend is not continuing. We notice the performance drops for a few countries. After manually checking such drops, we found that the model removes the wrong matches that are part of the ground truth data, which leads to a drop in the evaluation metrics. By adopting an iterative approach, we obtain a maximum improvement of 6 and 7% points in F1-score over Wikidata and DBpedia, respectively. Figure 4 displays the F1-scores for tag-to-class alignment after each iteration. We obtain a maximum increase of 4 and 8% points in the F1-score over Wikidata and DBpedia, respectively. We observe a similar trend as the entity alignment, such that the model performance increases up to the 3rd or 4th iteration. The increased number of aligned tag-class pairs provides more evidence for entity alignment.

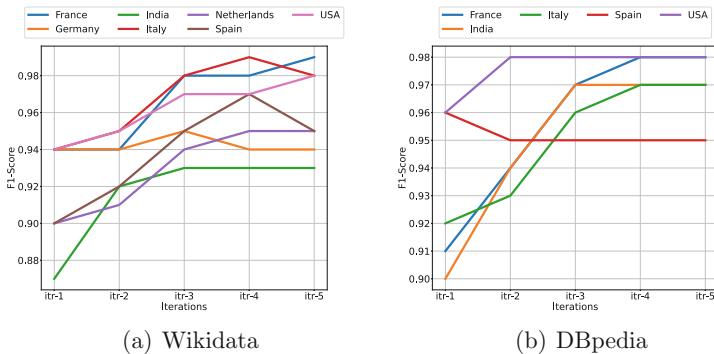


Fig. 3. Entity alignment performance: F1-scores for 1–5 iterations.

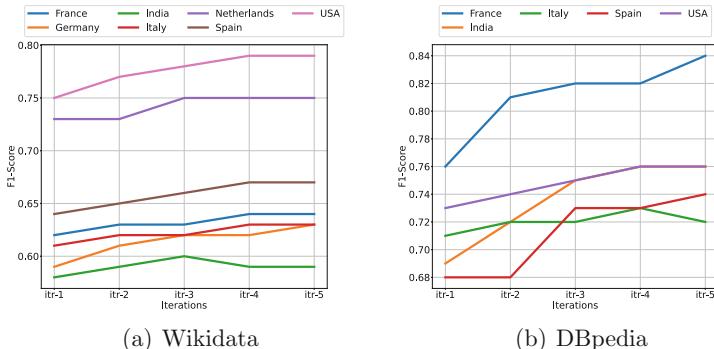


Fig. 4. Tag-to-class alignment performance: F1-scores for 1–5 iterations.

5.4 Alignment Threshold Tuning

We assess the importance of the alignment threshold th_a regarding the F1-score to select the appropriate value of th_a . Figure 5 depicts the F1-scores obtained after the third iteration for threshold values ranging between 0.50 and 0.90 with a gap of 0.1. Overall, the model performs well for all threshold values. Comparing the performance of different th_a values, the highest F1-score is achieved with a $th_a = 0.60$ for both KGs across all datasets. Therefore, in the experiments in other parts of this paper, we set th_a to 0.6.

5.5 Manual Assessment of New Links

We manually assess the quality of the links obtained on unseen data. We create the unseen dataset by considering the entities of Wikidata that are tagged with the country Germany and have a geo-coordinate, but are not present in the ground truth links. We randomly select 100 entities from all iterations and manually verify the correctness of the links. Out of 100 matches, we obtained 89 correct matches. We observe that 6 of the wrong matches are mostly located closer to each other or contained in one another. These entities contain similar property and tag values, making it difficult for the model to understand the difference. For example, Wikidata entity *Q1774543 (Klinger mühle)* is contained in OSM node *114219911 (Bessenbach)*. The lack of an English label also hinders the performance. Meanwhile, we observed that IGEA discovers new links between entities and corrects the previously wrong-linked entities. OSM node *1579461216 (Beuel-Ost)* has a Wikidata tag as *Q850834 (Beuel-Mitte)* but using IGEA, the correct Wikidata entity *Q850829 (Beuel-Ost)* has been linked to the OSM node. The performance of the unseen entities demonstrates the effectiveness of the proposed IGEA approach.

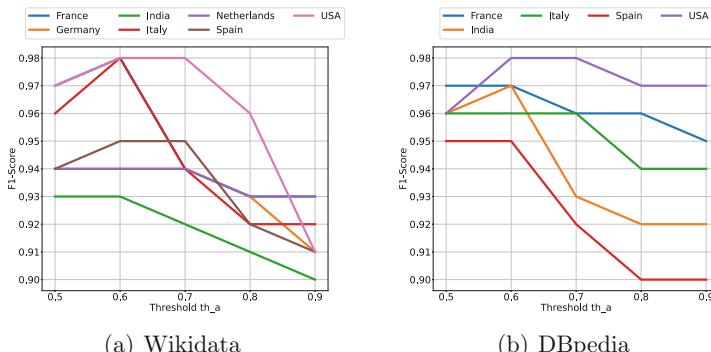


Fig. 5. Entity alignment performance in terms of F1-Score with different threshold values.

6 Related Work

This section discusses related work in geographic entity alignment, ontology alignment, and iterative learning.

Geographic entity alignment aims to align geographic entities across different geographic sources that refer to the same real-world object. In the past, approaches often relied on geographic distance and linguistic similarity between the labels of the entities [1, 13]. LIMES [20] relies on rules to rate the similarity between entities and uses these rules in a supervised model to predict the links. Tempelmeier et al. [21] proposed the OSM2KG algorithm – a machine-learning model to learn a latent representation of OSM nodes and align them with knowledge graphs. OSM2KG also uses KG features such as name, popularity, and entity type to produce more precise links. Recently, deep learning-based models have gained popularity for the task of entity alignment on tabular data. DeepMatcher [8] and HierMatcher [17] use an embedding-based deep learning approach for predicting the matches for tabular datasets. Peeters et al. [18] use contrastive learning with supervision to match entities in small tabular product datasets. In contrast, IGEA adopts the entire entity description, including KG properties and OSM tags, to enhance the linking performance.

Ontology and schema alignment refer to aligning elements such as classes, properties, and relations between ontologies and schemas. Such alignment can be performed at the element and structural levels. Many approaches have been proposed for tabular and relational data schema alignment and rely on the structural and linguistic similarity between elements [5, 12, 16, 26]. Lately, deep learning methods have also gained popularity for the task of schema alignment [2]. Due to the OSM schema heterogeneity and flatness, applying these methods to OSM data is difficult. Recently, Dsouza et al. [6] proposed the NCA model for OSM schema alignment with knowledge graphs using adversarial learning. We adopt NCA as part of the proposed IGEA approach.

Iterative learning utilizes the results of previous iterations in the following iterations to improve the performance of the overall task. In knowledge graphs, iterative learning is mainly adopted in reasoning and completion tasks. Many approaches exploit rule-based knowledge to generate knowledge graph embeddings iteratively. These embeddings are then used for tasks such as link prediction [11, 27]. Zhu et al. [28] developed a method for entity alignment across knowledge graphs by iteratively learning the joint low-dimensional semantic space to encode entities and relations. Wang et al. [24] proposed an embedding model for continual entity alignment in knowledge graphs based on latent entity representations and neighbors. In cross-lingual entity alignment, Xie et al. [25] created a graph attention-based model. The model iteratively and dynamically updates the attention score to obtain cross-KG knowledge. Unlike knowledge graphs, OSM does not have connectivity between entities. Therefore, the aforementioned methods are not applicable to OSM. In IGEA, we employ class and entity alignment iteratively to alleviate the data heterogeneity as well as annotation and interlinking sparsity to improve the results of the geographic entity and schema alignment.

7 Conclusion

In this paper, we presented IGEA – a novel iterative approach for geographic entity alignment based on cross-attention. IGEA overcomes the differences in entity representations between community-created geographic data sources and knowledge graphs by using a cross-attention-based model to align heterogeneous context information and predict identity links between geographic entities. By iterating schema and entity alignment, the IGEA approach alleviates the annotation and interlinking sparsity of geographic entities. Our evaluation results on real-world datasets demonstrate that IGEA is highly effective and outperforms the baselines by up to 18% points F1-score in terms of entity alignment. Moreover, we observe improvement in the results of tag-to-class alignment. We make our code publicly available to facilitate further research⁷.

Supplemental Material Statement: Sect. 4 provides details for baselines and datasets. Source code, instructions on data collection, and for repeating all experiments are available from GitHub (see footnote 7).

Acknowledgements. This work was partially funded by the DFG, German Research Foundation (“WorldKG”, 424985896), the Federal Ministry for Economic Affairs and Climate Action (BMWK), Germany (“ATTENTION!”, 01MJ22012C), and DAAD/BMBF, Germany (“KOALA”, 57600865).

References

1. Auer, S., Lehmann, J., Hellmann, S.: LinkedGeoData: adding a spatial dimension to the web of data. In: Bernstein, A., et al. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 731–746. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04930-9_46
2. Bento, A., Zouaq, A., Gagnon, M.: Ontology matching using convolutional neural networks. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp. 5648–5653. ELRA (2020)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguistics **5**, 135–146 (2017)
4. Chiu, J.P.C., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. Trans. Assoc. Comput. Linguistics **4**, 357–370 (2016)
5. Demidova, E., Oelze, I., Nejdl, W.: Aligning freebase with the YAGO ontology. In: 22nd ACM International Conference on Information and Knowledge Management, pp. 579–588 (2013)
6. Dsouza, A., Tempelmeier, N., Demidova, E.: Towards neural schema alignment for OpenStreetMap and knowledge graphs. In: Hotho, A., et al. (eds.) ISWC 2021. LNCS, vol. 12922, pp. 56–73. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88361-4_4
7. Dsouza, A., Tempelmeier, N., Yu, R., Gottschalk, S., Demidova, E.: Worldkg: a world-scale geographic knowledge graph. In: CIKM ’21: The 30th ACM International Conference on Information and Knowledge Management, pp. 4475–4484. ACM (2021)

⁷ <https://github.com/alishiba14/IGEA>.

8. Fu, C., Han, X., He, J., Sun, L.: Hierarchical matching network for heterogeneous entity resolution. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, pp. 3665–3671 (2020)
9. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)
10. Graves, A., Jaitly, N., Mohamed, A.: Hybrid speech recognition with deep bidirectional LSTM. In: Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 273–278 (2013)
11. Guo, S., Wang, Q., Wang, L., Wang, B., Guo, L.: Knowledge graph embedding with iterative guidance from soft rules. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence, pp. 4816–4823 (2018)
12. Jiménez-Ruiz, E., Agibetov, A., Chen, J., Samwald, M., Cross, V.: Dividing the ontology alignment task with semantic embeddings and logic-based modules. In: Proceedings of the 24th European Conference on Artificial Intelligence, pp. 784–791. FAIA, IOS Press (2020)
13. Karalis, N., Mandilaras, G., Koubarakis, M.: Extending the YAGO2 knowledge graph with precise geospatial knowledge. In: Ghidini, C., et al. (eds.) ISWC 2019. LNCS, vol. 11779, pp. 181–197. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30796-7_12
14. Lehmann, J., et al.: Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**(2), 167–195 (2015)
15. Li, P., et al.: Selfdoc: self-supervised document representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5652–5660 (2021)
16. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with cupid. In: Proceedings of the 27th International Conference on Very Large Data Bases, pp. 49–58. Morgan Kaufmann (2001)
17. Mudgal, S., et al.: Deep learning for entity matching: a design space exploration. In: Proceedings of the 2018 International Conference on Management of Data, pp. 19–34. ACM (2018)
18. Peeters, R., Bizer, C.: Supervised contrastive learning for product matching. In: Companion of the Web Conference 2022, pp. 248–251. ACM (2022)
19. Rebele, T., et al.: YAGO: a multilingual knowledge base from wikipedia, wordnet, and geonames. In: Proceedings of the 15th International Semantic Web Conference, pp. 177–185 (2016)
20. Sherif, M.A., Ngomo, A.N., Lehmann, J.: Wombat - a generalization approach for automatic link discovery. In: Proceedings of the 14th Extended Semantic Web Conference, pp. 103–119 (2017)
21. Tempelmeier, N., Demidova, E.: Linking OpenStreetMap with knowledge graphs - link discovery for schema-agnostic volunteered geographic information. Future Gener. Comput. Syst. **116**, 349–364 (2021)
22. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the Annual Conference on Neural Information Processing Systems, pp. 5998–6008 (2017)
23. Vrandecic, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
24. Wang, Y., et al.: Facing changes: continual entity alignment for growing knowledge graphs. In: Sattler, U., et al. The Semantic Web - ISWC 2022. ISWC 2022. LNCS, vol. 13489, pp. 196–213. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19433-7_12

25. Xie, Z., Zhu, R., Zhao, K., Liu, J., Zhou, G., Huang, J.X.: Dual gated graph attention networks with dynamic iterative training for cross-lingual entity alignment. ACM Trans. Inf. Syst. **40**(3), 44:1–44:30 (2022)
26. Zhang, S., Balog, K.: Web table extraction, retrieval, and augmentation: a survey. ACM Trans. Intell. Syst. Technol. **11**(2), 13:1–13:35 (2020)
27. Zhang, W., et al.: Iteratively learning embeddings and rules for knowledge graph reasoning. In: Proceedings of the World Wide Web Conference, pp. 2366–2377. ACM (2019)
28. Zhu, H., Xie, R., Liu, Z., Sun, M.: Iterative entity alignment via joint knowledge embeddings. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 4258–4264 (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

