Towards Neural Schema Alignment for OpenStreetMap and Knowledge Graphs

Alishiba Dsouza[1(✉)], Nicolas Tempelmeier[2], and Elena Demidova[1]

[1] Data Science and Intelligent Systems (DSIS), University of Bonn, Bonn, Germany
{dsouza,elena.demidova}@cs.uni-bonn.de
[2] L3S Research Center, Leibniz Universität Hannover, Hannover, Germany
tempelmeier@L3S.de

Abstract. OpenStreetMap (OSM) is one of the richest, openly available sources of volunteered geographic information. Although OSM includes various geographical entities, their descriptions are highly heterogeneous, incomplete, and do not follow any well-defined ontology. Knowledge graphs can potentially provide valuable semantic information to enrich OSM entities. However, interlinking OSM entities with knowledge graphs is inherently difficult due to the large, heterogeneous, ambiguous, and flat OSM schema and the annotation sparsity. This paper tackles the alignment of OSM tags with the corresponding knowledge graph classes holistically by jointly considering the schema and instance layers. We propose a novel neural architecture that capitalizes upon a shared latent space for tag-to-class alignment created using linked entities in OSM and knowledge graphs. Our experiments aligning OSM datasets for several countries with two of the most prominent openly available knowledge graphs, namely, Wikidata and DBpedia, demonstrate that the proposed approach outperforms the state-of-the-art schema alignment baselines by up to 37% points F1-score. The resulting alignment facilitates new semantic annotations for over 10 million OSM entities worldwide, which is over a 400% increase compared to the existing annotations.

Keywords: OpenStreetMap · Knowledge graph · Neural schema alignment

1 Introduction

OpenStreetMap (OSM) has evolved as a critical source of openly available geographic information globally, including rich data from 188 countries. This information is contributed by a large community, currently counting over 1.5 million volunteers. OSM captures a vast and continuously growing number of geographic entities, currently counting more than 6.8 billion [15]. The descriptions of OSM entities consist of heterogeneous key-value pairs, so-called *tags*, and include over 80 thousand distinct keys. OSM keys and tags do not possess machine-readable semantics, such that OSM data is not directly accessible for semantic applications. Whereas knowledge graphs (KGs) can provide precise semantics for geographic entities, large publicly available general-purpose KGs like Wikidata [30],

A. Hotho et al. (Eds.): ISWC 2021, LNCS 12922, pp. 56–73, 2021.
https://doi.org/10.1007/978-3-030-88361-4_4

DBpedia [2], YAGO [26], and specialized KGs like EventKG [10], and LinkedGeoData [25] lack coverage of geographic entities. For instance, in June 2021, 931,574 entities with tag `amenity=restaurant` were present in OSM, whereas Wikidata included only 4,391 entities for the equivalent class "restaurant".

An alignment of OSM and knowledge graphs at the schema level can make a wide variety of geographic entities in OSM accessible through semantic technologies and applications. The automatic suggestions of alignment candidates can help to create accurate schema mappings in human-in-the-loop applications. Furthermore, alignment models can help OSM volunteers to map geographic entities in OSM and annotate these entities with KG classes.

The problem of schema alignment between OSM and KGs is particularly challenging due to several factors, most prominently including the heterogeneous representations of types and properties of geographic entities via OSM tags, unclear tag semantics, the large scale and flatness of the OSM schema, and the sparseness of the existing links. OSM does not limit the usage of keys and tags by any strict schema and provides only a set of guidelines[1]. As a result, the types and properties of OSM entities are represented via a variety of tags that do not possess precise semantics. Consider an excerpt from the representations of the entity "Zugspitze" (mountain in Germany) in Wikidata and OSM:

Wikidata

Subject	Predicate	Object
$Q3375$	*label*	*Zugspitze*
$Q3375$	*coordinate*	$47°25'N, 10°59'E$
$Q3375$	*parentpeak*	$Q15127$
$Q3375$	*instance of*	*mountain*

OpenStreetMap

Key	Value
id	27384190
`name`	*Zugspitze*
`natural`	*peak*
`summit:cross`	*yes*

In Wikidata, an entity type is typically represented using the `instance of` property. In this example, the statement "Q3375 `instance of` mountain" indicates the type "mountain" of the entity "Q3375". In OpenStreetMap, the type "mountain" of the same entity is indicated by the tag `natural=peak`. As OSM lacks a counterpart of the `instance of` property, it is unclear which particular tag represents an entity type and which tags refer to other properties. Furthermore, multiple OSM tags can refer to the same semantic concept. Finally, whereas the OSM schema with over 80 thousand distinct keys is extensive, the alignment between OSM and knowledge graphs at the schema level is almost nonexistent. For instance, as of April 2021, Wikidata contained 585 alignments between its properties and OSM keys, corresponding to only 0.7% of the distinct OSM keys. Overall, the flatness, heterogeneity, ambiguity, and the large scale of OSM schema, along with a lack of links, make the alignment particularly challenging.

Existing approaches for schema alignment operate at the schema and instance level and consider the similarity of schema elements, structural similarity, and instance similarity. As OSM schema is flat, ontology alignment methods that utilize hierarchical structures, such as [13,17], are not applicable. A transformation of OSM data into a tabular or relational format leads to highly sparse tables with

[1] OSM "How to map a": https://wiki.openstreetmap.org/wiki/How_to_map_a.

numerous columns. Therefore, approaches to syntactic or instance-based alignment for relational or tabular data, such as e.g., [6,32], or syntactic matching of schema element names [28] cannot yield good results for matching OSM tags with KG classes.

This paper takes the first important step to align OSM and knowledge graphs at the schema level using a novel neural method. In particular, we tackle tag-to-class alignment, i.e., we aim to identify OSM tags that convey class information and map them to the corresponding classes in the Wikidata knowledge graph and the DBpedia ontology. We present the Neural Class Alignment (NCA) model - a novel instance-based neural approach that aligns OSM tags with the corresponding semantic classes in a knowledge graph. NCA builds upon a novel shared latent space that aligns OSM tags and KG concepts and facilitates a seamless translation between them. To the best of our knowledge, NCA is the first approach to align OSM and KGs at the schema level with a neural method.

Our contributions are as follows:

- We present NCA – a novel approach for class alignment for OSM and KGs.
- We propose a novel shared latent space that fuses feature spaces from knowledge graphs and OSM in a joint model, enabling simultaneous training of the schema alignment model on heterogeneous semantic and geographic sources.
- We develop a novel, effective algorithm to extract tag-to-class alignments from the resulting model.
- The results of our evaluation demonstrate that the proposed NCA approach is highly effective and outperforms the baselines by up to 37% points F1-score.
- As a result of the proposed NCA alignment method, we provide semantic annotations with Wikidata and DBpedia classes for over 10 million OSM entities. This result corresponds to an over 400% increase compared to currently existing annotations.
- We make our code and datasets publicly available and provide a manually annotated ground truth for the tag-to-class alignment of OSM tags with Wikidata and DBpedia classes[2].

2 Problem Statement

In this section, we formalize the problem definition. First, we formally define the concepts of an OSM corpus and a knowledge graph. An OSM corpus contains nodes representing geographic entities. Each node is annotated with an identifier, a location, and a set of key-value pairs known as tags.

Definition 1. *An* OSM corpus $\mathcal{C} = (N, T)$ *consists of a set of nodes* N *representing geographic entities, and a set of tags* T*. Each tag* $t \in T$ *is represented as a key-value pair, with the key* $k \in K$ *and a value* $v \in V$*:* $t = \langle k, v \rangle$*. A node* $n \in N$*,* $n = \langle i, l, T_n \rangle$ *is represented as a tuple containing an identifier* i*, a geographic location* l*, and a set of tags* $T_n \subset T$*.*

[2] GitHub repository: https://github.com/alishiba14/NCA-OSM-to-KGs.

A knowledge graph contains real-world entities, classes, properties, and relations.

Definition 2. *A* knowledge graph $\mathcal{KG} = (E, C, P, L, F)$ *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 triples* $F \subseteq E \times P \times (E \cup L)$*.*

The entities in E represent real-world entities and semantic classes. The properties in P represent relations connecting two entities, or an entity and a literal value. An entity in a KG can belong to one or multiple classes. An entity is typically linked to its class using the `rdf:type`, or an equivalent property.

Definition 3. *A class of the entity* $e \in E$ *in the knowledge graph* $\mathcal{KG} = (E, C, P, L, F)$ *is denoted as: class(e)* $= \{c \in C \mid (e, \texttt{rdf:type}, c) \in F\}$.

An OSM node and a KG entity referring to the same real-world geographic entity and connected via an identity link are denoted linked entities.

Definition 4. *A linked entity* $(n, e) \in E_L$ *is a pair of an OSM node* $n = \langle i, l, T_n \rangle$*,* $n \in N$*, and a knowledge graph entity* $e \in E$ *that corresponds to the same real-world entity. In a knowledge graph, a linked entity is typically represented using a* $(e, \texttt{owl:sameAs}, i)$ *triple, where* i *is the node identifier.* E_L *denotes the set of all linked entities in a knowledge graph.*

This paper tackles the alignment of tags that describe node types in an OSM corpus to equivalent classes in a knowledge graph.

Definition 5. ***Tag-to-class alignment:*** *Given a knowledge graph* $\mathcal{KG}$ *and an OSM corpus* $\mathcal{C}$*, find a set of pairs* $tag\_class \subseteq (T \times C)$ *of OSM tags* T *and the corresponding* $\mathcal{KG}$ *classes, such that for each pair* $(t, c) \in tag\_class$ *OSM nodes with the tag* t *belong to the class* c*.*

3 Neural Class Alignment Approach

An alignment of an OSM corpus with a knowledge graph can include several dimensions, such as entity linking, node classification (i.e., aligning OSM nodes with the corresponding semantic classes in a knowledge graph), as well as alignment of schema elements such as keys/tags and the corresponding semantic classes. The alignments in these dimensions can reinforce each other. For example, linking OSM nodes with knowledge graph entities and classifying OSM nodes into knowledge graph classes can lead to new schema-level alignments and vice versa. Our proposed NCA approach systematically exploits the existing identity links between OSM nodes and knowledge graph entities based on this intuition. NCA builds an auxiliary classification model and utilizes this model to align OSM tags with the corresponding classes in a knowledge graph ontology.

NCA is an unsupervised two-step approach for tag-to-class alignment. Figure 1 presents an overview of the proposed NCA architecture. First, we build

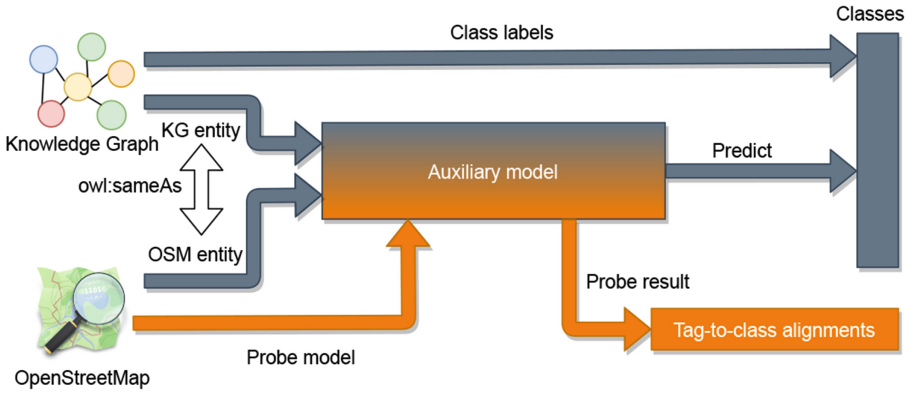


Fig. 1. Overview of the NCA architecture. The gray color indicates the first step (training of the auxiliary classification model). The orange color indicates the second step, i.e., the extraction of tag-to-class alignments. (Color figure online)

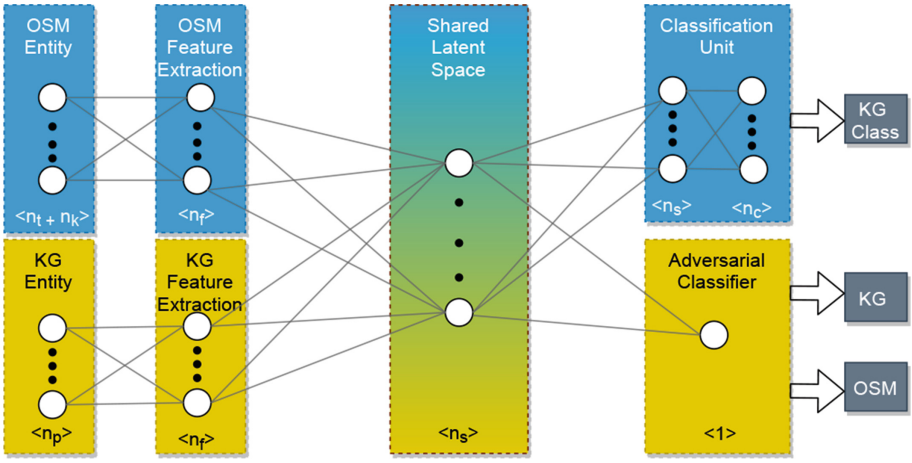


Fig. 2. The auxiliary classification model architecture. The blue color indicates the KG classification component, yellow marks the adversarial entity discrimination component. Parameters inside angular brackets denote the number of neurons in each layer, and lines denote the fully connected layers. (Color figure online)

an auxiliary neural classification model and train this model using linked entities in OSM and a KG. As a result, the model learns a novel shared latent space that aligns the feature spaces of OSM and a knowledge graph and implicitly captures tag-to-class alignments. Second, we systematically probe the resulting model to identify the captured alignments.

3.1 Auxiliary Neural Classification Model

In this step, we build a supervised auxiliary neural classification model for a dummy task of OSM node and KG entity classification. The model resulting from this step is later used for the tag-to-class alignment. Figure 2 presents the model architecture. The parameters n_t, n_k, n_p, n_c denote the number of OSM tags, number of OSM keys, number of KG properties, and number of KG classes, respectively. We experimentally select the number of neurons in the feature extraction layer (n_f) and the shared latent space layer (n_s). The auxiliary classification model architecture consists of several components described below.

OSM Node Representation. We represent an OSM node as a binary vector in an **O**-dimensional vector space. The space dimensions correspond to OSM tags or keys, and binary values represent whether the node includes the corresponding tag or key. The vector space dimensions serve as features for the classification model, such that we also refer to this space as the OSM feature space. To select the most descriptive tags to be included as dimensions in the OSM feature space, we filter out low-quality tags using OSM taginfo[3]. We include only the tags with an available description in the OSM wiki[4] having at least 50 occurrences within OSM. For tags with infrequent values (e.g., literals), we include only the keys as dimensions. We aim to align geographic concepts and not specific entities; thus, we do not include infrequent and node-specific values such as entity names or geographic coordinates in the representation. For instance, the concept of "mountain" is the same across different geographic regions, such that the geographic location of entities is not informative for the schema alignment.

KG Entity Representation. We represent a KG entity as a binary vector in a **V**-dimensional vector space. The space dimensions correspond to the KG properties. Binary values represent whether the entity includes the corresponding property. The vector space dimensions serve as features for the classification model, such that we also refer to this space as the KG feature space. To select the most descriptive properties to be included in the KG feature space, we rank the properties based on their selectivity concerning the class and the frequency of property usage (i.e., the number of statements in the KG that assign this property to an entity). Given a property p, we calculate its weight as: $weight(p, c) = n_{p,c} * log\frac{N}{c_p}$. Here, $n_{p,c}$ denotes the number of statements in which the property p is assigned to an entity of class c, N denotes the total number of classes in a knowledge graph, and c_p is the number of distinct classes that include the property p. For each class c, we select top-25 properties as features. These properties are included as dimensions in the KG feature space.

OSM & KG Feature Extraction. The KG and OSM feature representations serve as input to the specific fully connected feature extraction layers: OSM feature extraction and KG feature extraction. The purpose of these layers is to refine the vector representations obtained in the previous step.

[3] OSM taginfo: https://taginfo.openstreetmap.org/tags.
[4] OSM wiki: https://wiki.openstreetmap.org/wiki/.

Shared Latent Space & Adversarial Classifier. We introduce a novel *shared latent space* that fuses the initially disjoint feature spaces of OSM and KG such that entities from both data sources are represented in a joint space similarly. In addition to the training on OSM examples, shared latent space enables us to train our model on the KG examples. These examples provide the properties known to indicate class information [21]. The shared latent space component consists of a fully connected layer that receives the input from the OSM and KG feature extraction layers. Following recent domain adaption techniques [9], we use an adversarial classification layer to align latent representations of KG and OSM entities. The objective of the adversarial classifier is to discriminate whether the current training example is an OSM node or a KG entity, where the classification loss is measured as binary cross-entropy.

$$BinaryCrossEntropy = -\frac{1}{n}\sum_{i=1}^{n}[y_i \times \log(\hat{y}_i) + (1 - y_i) \times \log(1 - \hat{y}_i)],$$

where n is the total number of examples, y_i is the true class label, and $\hat{y}_i$ is the predicted class label. Intuitively, in a shared latent space, the classifier should not be able to distinguish whether a training example originates from OSM or a KG. To fuse the initially disjoint feature spaces, we reverse the gradients from the adversarial classification loss: $\mathcal{L}_{adverse} = -BinaryCrossEntropy_{adverse}$.

Classification Unit. To train the auxiliary classification model for the OSM nodes, we exploit linked entities. We label OSM nodes with semantic classes of equivalent KG entities. We use these class labels as supervision in the OSM node classification task. More formally, given a linked entity, $(n, e) \in E_L$, the training objective of the model is to predict $class(e)$ from n. Analogously, the training objective for a KG entity e is to predict the class label $class(e)$ of this entity.

We utilize a 2-layer feed-forward network as a classification model. In the last prediction layer of this network, each neuron corresponds to a class. As an entity can be assigned to multiple classes, we use a sigmoid activation function and a binary cross-entropy loss to achieve multi-label classification: $\mathcal{L}_{classifcation} = BinaryCrossEntropy_{classification}$. Finally, the joint loss function $\mathcal{L}$ of the network is given by $\mathcal{L} = \mathcal{L}_{classifcation} + \mathcal{L}_{adverse}$. In the training process, we alternate OSM and KG instances to avoid bias towards one data source.

3.2 Tag-to-Class Alignment

In this step, we systematically probe the trained auxiliary classification model to extract the tag-to-class alignment. The goal of this step is to obtain the corresponding KG class for a given OSM tag. Algorithm 1 details the extraction process. First, we load the pre-trained auxiliary model m (line 1) and initialize the result set (line 2). We then probe the model with a given list of OSM tags $\mathcal{T}$ (line 3). For a single tag $t \in \mathcal{T}$, we feed t to the OSM input layer of the auxiliary

Algorithm 1. Extract Tag-to-Class Alignment

`Input:`	m	Trained auxiliary model
	$\mathcal{T}$	List of OSM tags
	th_a	Alignment threshold
`Output:`	$align \subseteq (T \times C)$	Extracted alignment of tags and classes

```
1: load(m)
2: align ⇐ ∅
3: for all t ∈ T do
4:    forward_propagation(t, m)
5:    activations ⇐ extract_activations(m)
6:    for all a ∈ activations do
7:       if a > th_a then
8:          align ⇐ align ∪ {(t, class(a))}
9:       end if
10:   end for
11: end for
12: return align
```

model and compute the complete forward propagation of t within m (line 4). We then extract the activation of the neurons of the last layer of the classification model before the sigmoid nonlinearity (line 5). As the individual neurons in this layer directly correspond to KG classes, we expect that the activation of the specific neurons quantifies the likeliness that the tag t corresponds to the respective class. For each activation of a specific neuron a that is above the alignment threshold th_a (line 6–7), we extract the corresponding class c and add this class to the set of alignments (line 8). We determine the threshold value experimentally, as described later in Sect. 5.3. As an OSM tag can have multiple corresponding classes, we opt for all matches above the threshold value. Finally, the resulting set *align* constitutes the inferred tag-to-class alignments.

3.3 Illustrative Example

We illustrate the proposed NCA approach at the example of the "Zugspitze" mountain introduced in Sect. 1. We create the representation of the Wikidata object "Q3375" in the KG feature space by creating a binary vector that has ones in the dimensions that correspond to the properties that this entity contains, such as, `label, coordinate, parentpeak`, and zeros otherwise. Note that the `instance of` predicate is not included in the feature space, as this predicate represents the class label. Similarly, we encode the OSM node with the id "27384190" in the OSM feature space by creating a vector that includes `name, natural=peak, summit:cross` as ones, and zeros in all other dimensions. As described above, we use frequent key-value pairs such as `natural=peak` as features, whereas for the infrequent key-value pairs, such as `name=Zugspitze`, we use only the key (i.e., `name`) as a feature. The KG and OSM features spaces

are then aligned in the shared latent space. To form this space, we train the auxiliary classification model that learns to output the correct class labels, such as "mountain". In the last prediction layer of this model, each neuron corresponds to a class. After the training is completed, we probe the classification model with a single tag, such as `natural=peak`. The activation of the neurons in the prediction layer corresponds to the predicted tag-to-class mapping. We output all classes with the activation values above the threshold th_a (here: "mountain").

4 Evaluation Setup

This section introduces the evaluation setup regarding datasets, ground truth generation, baselines, and evaluation metrics. All experiments were conducted on an AMD Opteron 8439 SE processor @ 2.7 GHz and 252 GB of memory, whereas the execution of NCA required up to 16 GB of memory only.

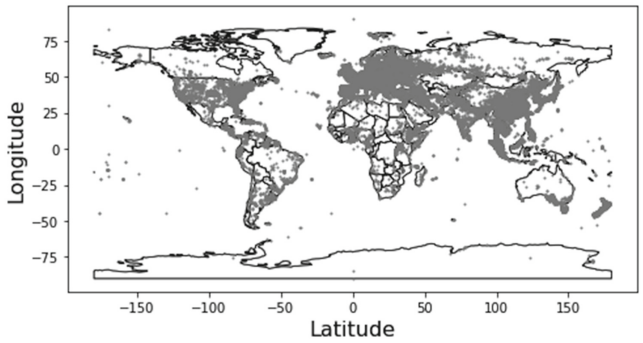


Fig. 3. OSM and Wikidata linked entities located on a world map.

4.1 Datasets

We carry out our experiments on OSM, Wikidata [30], and DBpedia [2] datasets.

Knowledge Graphs: A sufficient number of linked entities and distinct classes is essential to train the proposed neural model and achieve a meaningful schema alignment. As illustrated in Fig. 3, OSM to Wikidata links are highly frequent in the European region. We systematically rank European countries according to the number of linked entities between OSM and knowledge graphs. We choose the top-4 countries having at least ten distinct classes in the linked entity set. Based on these criteria, we select the Wikidata datasets for France, Germany, Great Britain, and Russia as well as the DBpedia datasets for France, Germany, Great Britain, and Spain. Although over 100,000 entity links between Russian DBpedia and OSM exist, most entities belong to only two classes. Hence, we omit Russian DBpedia from our analysis. Additionally, to understand the effect of NCA in other parts of the world, we select the USA and Australia with a

moderate amount of KG links. In our experiments, we consider Wikidata and DBpedia snapshots from March 2021. We collect the data from knowledge graphs by querying their SPARQL endpoints. We only consider geographic entities, i.e., the entities with valid geographic coordinates.

OpenStreetMap: We extract OSM data for France, Germany, Great Britain, Spain, Russia, the USA, and Australia. To facilitate evaluation, we only consider OSM nodes which include links to knowledge graphs. The number of entities assigned to specific knowledge graph classes follows a power-law distribution. We select the classes with more than 100 entities (i.e., 3% of classes in Wikidata) to facilitate model training. Note that some KG entities are linked to more than one OSM node, such that the number of nodes and entities in the dataset differ.

4.2 Ground Truth Creation

For Wikidata, we start the creation of our ground truth based on the "OpenStreetMap tag or key" Wikidata property[5]. This property provides a link between a Wikidata class and the corresponding OSM tag. However, this dataset is incomplete and lacks some language-specific classes as well as superclass and subclass relationships based on our manual analysis. We manually extended the ground truth by checking all possible matches obtained by the proposed NCA approach and all baseline models used in the evaluation. We added all correct matches to our ground truth. For DBpedia, we constructed the ground truth manually by labeling all combinations ($T \times C$) of OSM tags t and $\mathcal{KG}$ classes C in our dataset. For both KGs, we consider region-specific matches ("Ortsteil" vs. "District") and subclass/superclass relations (e.g., "locality" vs. "city/village").

4.3 Baselines

The schema alignment task of OSM and KG has not been addressed before, such that no task-specific baseline exists. For evaluation, we choose the state-of-the-art baselines from schema alignment for tabular data (Cupid [13], EmbDI [5], Similarity Flooding [14]), which is the closest representation to the OSM flat schema structure. Furthermore, we evaluate string similarity using Levenshtein distance, word embeddings-based cosine similarity, and SD-Type [21] - an established approach for type inference. To fit our data to the baselines, we convert our OSM (source) data and KG (target) data into a tabular format. For OSM, we use the tags and keys as columns and convert each node into a row. Similarly, for KGs, the properties and classes are converted into columns, and the entities form the rows. We evaluate our proposed method against the following baselines:

Cupid: Cupid [13] matches schema elements based on element names, structure, and data types. Cupid is a 2-phase approach. The first phase calculates the lexicographic similarity of names and data types. The second phase matches

[5] Wikidata "OpenStreetMap tag or key" property: https://www.wikidata.org/wiki/Property:P1282.

elements using the structural similarity based on the element proximity in the ontology hierarchy. As the OSM schema is flat, we consider a flat hierarchy, where the OSM table is the root and all columns are child nodes. The final Cupid score is the average similarity between the two phases.

Levenshtein Distance (LD): The Levenshtein distance (edit distance) is a string-based similarity measure used to match ontology elements lexicographically. The Levenshtein distance between two element names is calculated as the minimal number of edits needed to transform one element name to obtain the other. The modifications include addition, deletion, or replacement of characters [28]. We calculate the Levenshtein distance between all pairs of class names and tags and accept pairs with a distance lower than the threshold $th_l \in [0, 1]$.

EmbDi: EmbDi [5] is an algorithm for schema alignment and entity resolution. The algorithm maps table rows to a directed graph based on rows, columns, and cell values. EmbDi infers column embeddings by performing random walks on the graph. The random walks form sentences that constitute an input to a Word2Vec model. Finally, the similarity of the two columns is measured as the cosine similarity of the respective embeddings.

Similarity Flooding (SF): Similarity Flooding [14] transforms a data table into a directed labeled graph in which the nodes represent table columns. The weights of graph edges represent the node similarity, initialized using string similarity of the column names. The algorithm refines the weights by iteratively propagating similarity values along the edges. Each pair of nodes connected with a similarity value above the matching threshold forms an alignment.

SD-Type (SD): SD-Type [21] is an established approach for type inference. While SD-type was originally proposed to infer instance types based on conditional probabilities, we transfer the idea to infer class types. We calculate the conditional probability of a tag t given a class c as follows: $p(c|t) = \frac{\sum(t \bigcap c)}{\sum t}$. We accept all the matches with the probability values above threshold $th_l \in [0, 1]$.

Word Embedding Based Cosine Similarity (WECS): We use pre-trained word embeddings[6] trained using fastText [4] with 300 dimensions to obtain the word vectors of tag and class names. We calculate the cosine similarity between the word vectors of each tag-class pair. We accept all pairs with cosine similarity above the threshold $th_l \in [0, 1]$ as a match.

For LD, SD and WECS, we apply an exhaustive grid search to optimize the value of th_l for each dataset and report the highest resulting F1-scores. For the Cupid, EmbDi, and SF baseline implementation, we use the source code from the delftdata GitHub repository[7].

4.4 Metrics

The standard evaluation metrics for schema alignment are precision, recall, and F1-score computed against a reference alignment (i.e., ground truth). We eval-

[6] https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.bin.gz.
[7] Delftdata GitHub repository: https://github.com/delftdata/valentine.

uate the mappings as pairs, where each pair consists of one tag and one class (tag-to-class alignment). **Precision** is the fraction of correctly identified pairs among all identified pairs. **Recall** is the fraction of correctly identified pairs among all pairs in the reference alignment. **F1-score** is the harmonic mean of recall and precision. We consider the F1-score to be the most relevant metric since it reflects both precision and recall.

5 Evaluation

The evaluation aims to assess the performance of the proposed NCA approach for tag-to-class alignment in terms of precision, recall, and F1-score. Furthermore, we aim to analyze the influence of the confidence threshold and the impact of the shared latent space on the alignment performance. Note that we do not evaluate the artificial auxiliary classification task. Instead, we evaluate the utility of the auxiliary model in the overall schema alignment task. We train and evaluate the models for each country and knowledge graph separately.

5.1 Tag-to-Class Alignment Performance

Table 1 and 2 summarize the performance results of the baselines and our proposed NCA approach with respect to precision, recall and F1-score for tag-to-class alignment of OSM tags to Wikidata and DBpedia classes, respectively. As we can observe, the proposed NCA approach outperforms the baselines in terms of F1-score on all datasets. On Wikidata, we achieve up to 13% points F1-score improvement and ten percentage points on average compared to the best baseline. On DBpedia, we achieve up to 37% points F1-score improvement and 21% points on average. As OSM lacks a hierarchical structure, limiting structural comparison, most of the applicable baselines build on the name comparison. Here, the heterogeneity of OSM tags limits the precision of the baselines substantially. SD-Type obtains the highest F1-score amongst baselines. NCA uses the property, tags, and keys information from the shared latent space and achieves higher performance than the best performing SD-Type baseline. For other baselines, the absolute values achieved are relatively low. SF, WECS, and EmbDI obtain only low similarity values, resulting in low precision. An increase of the confidence threshold for these baselines leads to zero matches. The tag-class pairs vary significantly in terms of linguistic and semantic similarities. The correct pairs obtained using WECS do not obtain sufficiently high scores to discriminate from the wrong matches, making WECS one of the weakest baselines.

We observe performance variations across countries and knowledge graphs, with Australian Wikidata and French DBpedia achieving the highest F1-scores compared to other countries. These variations can be explained by the differences in the dataset characteristics, including the number of links, entities per class, and unique tags and classes per country. These characteristics vary significantly across the datasets. Furthermore, the number of classes per entity varies. On average, Wikidata indicates one class per entity (i.e., the most specific class).

Table 1. Tag-to-class alignment performance for OSM tags to Wikidata classes.

Name	France			Germany			Great Britain			Russia			USA			Australia			Average		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Cupid	0.06	1.00	0.12	0.03	0.70	0.06	0.07	1.00	0.14	0.08	0.80	0.15	0.06	1.00	0.11	0.25	1.00	0.38	0.09	0.91	0.16
LD	0.45	0.28	0.35	0.65	0.34	0.44	0.54	0.37	0.44	0.64	0.34	0.45	0.39	0.37	0.38	0.31	0.41	0.36	0.49	0.35	0.40
EmbDi	0.03	1.00	0.06	0.02	1.00	0.03	0.04	1.00	0.06	0.02	1.00	0.03	0.01	1.00	0.03	0.08	0.91	0.15	0.05	0.98	0.06
SF	0.03	1.00	0.06	0.02	1.00	0.03	0.01	1.00	0.03	0.02	1.00	0.03	0.01	1.00	0.03	0.08	1.00	0.16	0.04	1.00	0.06
WECS	0.35	0.09	0.14	0.23	0.16	0.19	0.10	0.28	0.14	0.25	0.29	0.26	0.23	0.06	0.09	0.13	0.53	0.21	0.22	0.23	0.16
SD	0.73	0.55	0.63	0.72	0.36	0.48	0.88	0.33	0.49	0.45	0.45	0.48	0.84	0.40	0.54	0.95	0.55	0.70	0.76	0.44	0.55
NCA	0.63	0.66	**0.65**	0.59	0.65	**0.61**	0.71	0.56	**0.63**	0.64	0.51	**0.58**	0.79	0.61	**0.69**	0.85	0.78	**0.82**	0.70	0.63	**0.66**

Table 2. Tag-to-class alignment performance for OSM tags to DBpedia classes.

Name	France			Germany			Great Britain			Spain			USA			Australia			Average		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Cupid	0.32	1.00	0.48	0.18	1.00	0.31	0.41	1.00	0.58	0.44	1.00	0.63	0.10	1.00	0.17	0.48	1.00	0.65	0.32	1.00	0.47
LD	0.31	0.57	0.41	0.32	0.37	0.34	0.73	0.46	0.57	0.34	0.94	0.50	0.42	0.97	0.59	0.58	0.62	0.60	0.45	0.65	0.50
EmbDi	0.16	1.00	0.28	0.09	1.00	0.17	0.29	1.00	0.45	0.24	1.00	0.38	0.33	1.00	0.51	0.32	1.00	0.50	0.24	1.00	0.38
SF	0.14	1.00	0.27	0.10	1.00	0.18	0.27	1.00	0.42	0.24	1.00	0.39	0.33	1.00	0.50	0.30	1.00	0.46	0.23	1.00	0.37
WECS	0.30	65	0.41	0.16	0.97	0.28	0.22	0.96	0.36	0.38	0.67	0.49	0.41	0.95	0.57	0.45	0.66	0.53	0.32	0.81	0.44
SD	0.92	0.57	0.70	0.34	0.98	0.50	0.57	0.88	0.69	0.83	0.58	0.69	0.70	0.47	0.58	0.95	0.55	0.70	0.71	0.67	0.64
NCA	0.95	0.90	**0.92**	0.96	0.79	**0.87**	0.81	0.84	**0.83**	1.00	0.84	**0.91**	0.70	0.70	**0.70**	0.95	0.76	**0.85**	0.90	0.81	**0.85**

Table 3. Example tag-to-class alignments obtained using the NCA approach.

Wikidata: France	Germany	Great Britain	Russia	USA	Australia
amenity=bicycle_rental: bicycle-sharing station	amenity=cinema: movie theater	railway=station: railway station	station=subway: metro station	landuse=reservoir: reservoir	amenity=library: public library
DBpedia: France	Germany	Great Britain	Spain	USA	Australia
railway=station: Place	place=municipality: Place	place=hamlet: Place	railway=station: ArchitecturalStructure	man_made=lighthouse: Location	public_transport=station: Infrastructure

In contrast, DBpedia indicates three classes per entity (i.e., the specialized and more generic classes at the higher levels of the DBpedia ontology). This property makes the model trained on the DBpedia knowledge graph more confident regarding the generic classes, such that generic classes obtain higher F1-scores than the specialized classes. Our observations indicate that it is desirable to obtain more training examples that align entities with more specific classes, such as in the Wikidata dataset. Table 3 illustrates the most confident tag-to-class alignments in terms of the obtained model activations using the NCA approach. As discussed above, Wikidata alignments with high confidence scores are more specific than those obtained on DBpedia.

5.2 Influence of the Shared Latent Space

Table 4 summarizes the performance of the proposed NCA approach and NCA without the shared latent space for tag-to-class alignment of OSM with Wikidata and DBpedia, respectively. We observe that the shared latent space helps to achieve an increase in F1-score of 34% points and 11% points for Wikidata and DBpedia, respectively. Compared to the Wikidata datasets, we observe smaller improvements on DBpedia datasets. DBpedia has an imbalance between the tags

Table 4. Tag-to-class alignment performance for Wikidata and DBpedia.

Approach	Avg. Wikidata			Avg. DBpedia		
	Precision	Recall	F1	Precision	Recall	F1
NCA w/o shared latent space	0.48	0.25	0.32	0.65	0.88	0.74
NCA	0.70	0.63	**0.66**	0.90	0.81	**0.85**

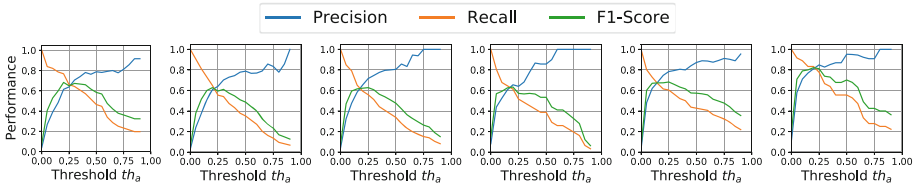
and classes, resulting in many-to-one alignments between tags and classes, where one class corresponds to several tags. For example, in all DBpedia datasets, the *place* and *populatedPlace* are frequently occurring classes for various tags such as `tourism=museum, place=village, place=town`. In such a case, DBpedia properties add less specific information to the matching process. Furthermore, we observe a high F1-score of the proposed NCA approach without the shared latent space on the DBpedia dataset. Intuitively, further improving these high scores is more difficult than improving the comparably low scores on Wikidata (e.g., 0.32 F1-score on Wikidata). In summary, the shared latent space improves the performance, with the highest improvements on Wikidata.

5.3 Confidence Threshold Tuning

We evaluate the influence of the confidence threshold value th_a on the precision, recall, and F1-score. The threshold th_a indicates the minimum similarity at which we align a tag to a class. Figure 4 and 5 present the alignment performance with respect to th_a for Wikidata and DBpedia. As expected, we observe a general trade-off between precision and recall, whereas higher values of th_a result in higher precision and lower recall. We select the confidence threshold of $th_a = 0.25$ and $th_a = 0.4$ for Wikidata and DBpedia, respectively, as these values allow balancing precision and recall. The threshold can be tuned for specific regions.

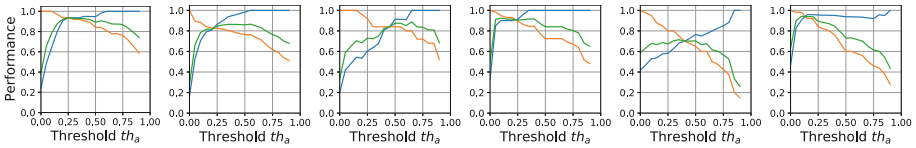
5.4 Alignment Impact

To assess the impact of NCA, we compare the number of OSM entities that can be annotated with semantic classes using the alignment discovery by NCA



(a) France (b) Germany (c) GB (d) Russia (e) USA (f) Australia

Fig. 4. Precision, recall, and F1-score vs. the confidence threshold for Wikidata. (Color figure online)



(a) France (b) Germany (c) GB (d) Spain (e) USA (f) Australia

Fig. 5. Precision, recall, and F1-score vs. the confidence threshold for DBpedia. (Color figure online)

with the number of entities that are linked to a KG in the currently existing datasets. For Wikidata, we observe 2,004,510 linked OSM entities and 10,163,762 entities annotated with semantic classes using NCA. This result corresponds to an increase of 407.04% of entities with semantic class annotations. For DBpedia, we observe 1,396,378 linked OSM entities and 8,301,450 entities annotated with semantic classes using NCA. This result corresponds to an increase of 494.5% of entities with semantic class annotations. We provide the resulting annotations as a part of the WorldKG knowledge graph[8].

6 Related Work

This work is related to ontology alignment, alignment of tabular data, feature space alignment, and link discovery.

Ontology Alignment. Ontology alignment (also ontology matching) aims to establish correspondences between the elements of different ontologies. The efforts to interlink open semantic datasets and benchmark ontology alignment approaches have been driven by the W3C SWEO Linking Open Data community project[9] and the Ontology Alignment Evaluation Initiative (OAEI)[10] [1]. Ontology alignment is conducted at the element-level and structure-level [20]. The element-level alignment typically uses natural language descriptions of the ontology elements, such as labels and definitions. Element-level alignment adopts string similarity metrics such as, e.g., edit distance. Structure-level alignment exploits the similarity of the neighboring ontology elements, including the taxonomy structure, as well as shared instances [17]. Element-level and structure-level alignment have also been adopted to align ontologies with relational data [6] and tabular data [32]. Jiménez-Ruiz et al. [11] divided the alignment task into independent, smaller sub-tasks, aiming to scale up to very large ontologies. In machine learning approaches, such as the GLUE architecture [7], semantic mappings are learned in a semi-automatic way. In [19], a matching system integrates string-based and semantic similarity features. Recently, more complex

[8] WorldKG knowledge graph: http://www.worldkg.org.
[9] https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.
[10] OAEI evaluation campaigns: http://oaei.ontologymatching.org.

approaches using deep neural networks have been proposed for ontology alignment and schema matching [3,22,31]. The lack of a well-defined ontology in OSM hinders the application of ontology alignment approaches. In contrast, the instance-based NCA approach enables an effective alignment of tags to classes.

Tabular Data Alignment. Another branch of research investigated the schema alignment of tabular data [23]. EmbDi [5] approach uses random walks and embeddings to find similarities between schema elements. Cupid [13] matches schema elements based on element names, structure, and data types. Similarity Flooding [14] transforms a table into a directed labeled graph in which nodes represent columns to compute similarity values iteratively. We employ the EmbDi, Cupid, and Similarity Flooding algorithms as baselines for our evaluation. Although the conversion of OSM key-value-based data into a tabular form is possible in principle, the resulting tables are highly sparse. Therefore, as seen in Sect. 4.3, tabular data alignment approaches do not perform well on the alignment task addressed in this work.

Feature Space Alignment. Recently, various studies investigated the alignment of feature spaces extracted from different data sources. Application domains include computer vision [8] and machine translation [12]. Ganin et al. [9] proposed a neural domain adaptation algorithm that considers labeled data from a source domain and unlabeled data from a target domain. While this approach was originally used to align similar but different distributions of feature spaces, we adopt the gradient reversal layer proposed in [9] to fuse information from the disjoint features spaces of OSM and KGs, not attempted previously.

Link Discovery. Link Discovery is the task of identifying semantically equivalent resources in different data sources [16]. Nentwig et al. [16] provide a recent survey of link discovery frameworks with prominent examples, including Silk [29] and LIMES [18]. In particular, the Wombat algorithm, integrated within the LIMES framework [24], is a state-of-the-art approach for link discovery in knowledge graphs. Specialized approaches [27] focus on link discovery between OSM and knowledge graphs. We build on existing links between OSM and knowledge graphs to align knowledge graph classes to OSM tags in this work.

7 Conclusion

In this paper, we presented NCA – the first neural approach for tag-to-class alignment between OpenStreetMap and knowledge graphs. We proposed a novel shared latent space that seamlessly fuses features from knowledge graphs and OSM in a joint model and makes them simultaneously accessible for the schema alignment. Our model builds this space as the core part of neural architecture, incorporating an auxiliary classification model and an adversarial component. Furthermore, we proposed an effective algorithm that extracts tag-to-class alignments from the resulting shared latent space with high precision. Our evaluation results demonstrate that NCA is highly effective and outperforms the baselines by up to 37% points F1-score. We make our code and manually annotated ground

truth data publicly available to facilitate further research. We believe that NCA is applicable to other geographic datasets having similar data structure as OSM; we leave such applications to future work.

Acknowledgements. This work was partially funded by DFG, German Research Foundation ("WorldKG", DE 2299/2-1), BMBF, Germany ("Simple-ML", 01IS18054) and BMWi, Germany ("d-E-mand", 01ME19009B).

References

1. Algergawy, A., et al.: Results of the ontology alignment evaluation initiative 2019. In: OM-2019. CEUR Workshop Proceedings, vol. 2536, pp. 46–85 (2019)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
3. Bento, A., Zouaq, A., Gagnon, M.: Ontology matching using convolutional neural networks. In: LREC 2020, pp. 5648–5653. ELRA (2020)
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
5. Cappuzzo, R., Papotti, P., Thirumuruganathan, S.: Creating embeddings of heterogeneous relational datasets for data integration tasks. In: SIGMOD 2020, pp. 1335–1349. ACM (2020)
6. Demidova, E., Oelze, I., Nejdl, W.: Aligning freebase with the YAGO ontology. In: CIKM 2013, pp. 579–588. ACM (2013)
7. Doan, A., Madhavan, J., Domingos, P.M., Halevy, A.Y.: Ontology matching: a machine learning approach. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, pp. 385–404. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24750-0_19
8. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV 2013. IEEE (2013)
9. Ganin, Y., et al.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**, 59:1–59:35 (2016)
10. Gottschalk, S., Demidova, E.: EventKG - the hub of event knowledge on the web - and biographical timeline generation. Semantic Web **10**(6), 1039–1070 (2019)
11. Jiménez-Ruiz, E., Agibetov, A., Chen, J., Samwald, M., Cross, V.: Dividing the ontology alignment task with semantic embeddings and logic-based modules. In: ECAI 2020. FAIA, vol. 325, pp. 784–791. IOS Press (2020)
12. Lample, G., Conneau, A., Ranzato, M., Denoyer, L., Jégou, H.: Word translation without parallel data. In: ICLR 2018. OpenReview.net (2018)
13. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with cupid. In: VLDB 2001, pp. 49–58. Morgan Kaufmann (2001)
14. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In: ICDE 2002 (2002)
15. Neis, P.: OSMstats. https://osmstats.neis-one.org/. Accessed 10 Apr 2021
16. Nentwig, M., Hartung, M., Ngomo, A.N., Rahm, E.: A survey of current link discovery frameworks. Semantic Web **8**(3), 419–436 (2017)

17. Ngo, D.H., Bellahsene, Z., Todorov, K.: Opening the black box of ontology matching. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) ESWC 2013. LNCS, vol. 7882, pp. 16–30. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38288-8_2
18. Ngomo, A.N., Auer, S.: LIMES - a time-efficient approach for large-scale link discovery on the web of data. In: IJCAI 2011, pp. 2312–2317. IJCAI/AAAI (2011)
19. Nkisi-Orji, I., Wiratunga, N., Massie, S., Hui, K., Heaven, R.: Ontology alignment based on word embedding and random forest classification. In: ECML PKDD (2018)
20. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: a literature review. Expert Syst. Appl. **42**(2), 949–971 (2015)
21. Paulheim, H., Bizer, C.: Type inference on noisy RDF data. In: ISWC 2013 (2013)
22. Qiu, L., Yu, J., Pu, Q., Xiang, C.: Knowledge entity learning and representation for ontology matching based on deep neural networks. Clust. Comput. **20**, 969–977 (2017)
23. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. **10**(4), 334–350 (2001)
24. Sherif, M.A., Ngonga Ngomo, A.-C., Lehmann, J.: WOMBAT – a generalization approach for automatic link discovery. In: Blomqvist, E., Maynard, D., Gangemi, A., Hoekstra, R., Hitzler, P., Hartig, O. (eds.) ESWC 2017. LNCS, vol. 10249, pp. 103–119. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58068-5_7
25. Stadler, C., Lehmann, J., Höffner, K., Auer, S.: LinkedGeoData: a core for a web of spatial open data. Semantic Web **3**(4), 333–354 (2012)
26. Pellissier Tanon, T., Weikum, G., Suchanek, F.: YAGO 4: a reason-able knowledge base. In: Harth, A., et al. (eds.) ESWC 2020. LNCS, vol. 12123, pp. 583–596. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49461-2_34
27. 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)
28. Unal, O., Afsarmanesh, H.: Using linguistic techniques for schema matching. In: ICSOFT 2006, pp. 115–120. INSTICC Press (2006)
29. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Silk - A link discovery framework for the web of data. In: LDOW 2009. CEUR, vol. 538. CEUR-WS.org (2009)
30. Vrandecic, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
31. Xiang, C., Jiang, T., Chang, B., Sui, Z.: ERSOM: a structural ontology matching approach using automatically learned entity representation. In: EMNLP (2015)
32. Zhang, S., Balog, K.: Web table extraction, retrieval, and augmentation: a survey. ACM Trans. Intell. Syst. Technol. **11**(2), 13:1–13:35 (2020)