Journal on Data Semantics (2020) 9:151–177
https://doi.org/10.1007/s13740-020-00117-4

ORIGINAL ARTICLE

Recommending Geo-semantically Related Classes for Link Discovery

Vasilis Kopsachilis[1] · Michail Vaitis[1] · Nikos Mamoulis[2] · Dimitris Kotzinos[3,4,5]

Received: 11 November 2019 / Revised: 30 November 2020 / Accepted: 2 December 2020 / Published online: 7 January 2021

Abstract
The growth of Web of Data led to the development of dataset recommendation methodologies, which automate the discovery of datasets that may contain same or related instances (i.e., objects), in order to be used as input for several tasks including Link Discovery. The recommendation process takes as input one dataset (or any tripleset) and proposes other datasets which are the most likely to contain related instances. Existing recommenders determine the relevance between datasets by comparing their textual and structural similarity or by examining existing links among them. In this paper, we determine relevancy by comparing the geospatial relatedness of triplesets containing instances belonging to spatial classes (that is, classes containing instances whose locations are georeferenced by point geometries) based on the hypothesis that pairs of classes whose instances present similar spatial distribution are likely to contain semantically related instances. The proposed methodology builds summaries that capture the spatial distribution of classes. It utilizes the summaries, first, to rule out irrelevant (to an input class) classes by applying spatial filters and, then, to rank the remaining classes by applying a geospatial relatedness measure, so as the top ranked classes are more probable to contain related instances. The methodology's evaluation contains an exploration of Web of Data spatial classes characteristics and a discussion of the experiment results that validate our hypothesis. We show that the spatial filtering reduces effectively and efficiently up to 99% the search space for relevant classes in Web of Data and that the proposed geospatial relatedness measures generate ranked lists of recommended classes with 62% mean average precision, approximately 35% higher than simple baselines.

Keywords Web of Data · Spatial information · Dataset recommendation · Geo-semantic relevance

1 Introduction

Over the last years, data providers have been publishing their data according to the Linked Data principles [5] weaving the Web of Data (WoD), a global data space where entities across the web are more discoverable and easier reusable [15]. A fundamental prerequisite for the realization of the WoD is the establishment of links between instances (dispersed across different datasets) for which some relation exists (e.g., using *sameAs* links for instances that represent the same real world object or *seeAlso* links for instances that provide additional information to a given instance). Toward the goal of link establishment, data providers are suggested to apply to their datasets *Link Discovery* methodologies, implemented in tools such as SILK [44] or LIMES [30]. Link Discovery refers to the process of identifying and interlinking pairs of instances between two given triplesets for which a relation holds [28]. A preprocessing step requires the specification of the triplesets that will be used as input in the Link Discovery task.

Linked Data practitioners may be unaware of triplesets that contain related instances or may want to explore WoD to link their data with additional resources. For this purpose, they, typically, look up for relevant datasets by manually examining the LOD cloud diagram,[1] which provides an overview of the datasets domain and connectivity, or by

✉ Vasilis Kopsachilis
vkopsachilis@geo.aegean.gr

Dimitris Kotzinos
Dimitrios.Kotzinos@cyu.fr

[1] Department of Geography, University of the Aegean, Mytilene, Greece

[2] Department of Computer Science and Engineering, University of Ioannina, Ioannina, Greece

[3] University of Paris-Seine, University of Cergy-Pontoise, Pontoise, France

[4] ETIS Lab, CY Cergy Paris University, Pontoise, France

[5] Lab. ETIS UMR 8051, CY Cergy Paris University, ENSEA, CNRS, F-95302, Pontoise, France

[1] https://lod-cloud.net/.

exploring dataset catalogs, such as datahub.io,[2] which preserve user submitted dataset's metadata. However, WoD is large: in 2019, the LOD cloud diagram[3] was including 1234 datasets, and LODStats[4] (in order to generate Web of Data statistics) parsed about 3000 datasets containing approximately 50 million entities in total. Additionally, is continuously expanding: during the period 2011–2017 the number of the LOD cloud datasets increased by 294% [35], thus turning the task of manual searching for relevant datasets into a challenging one. Consequently, data providers tend to link their data with well-known datasets (such as DBpedia or Geonames) and ignore less popular datasets that may also contain related instances [3,27,31]. As Leme et al. [20] points, linkage with popular datasets is favored because of two main reasons: the difficulty in finding relevant datasets and the strenuous task of discovering mappings between datasets. Works about WoD connectivity status reveal that 44% of datasets do not contain links to other datasets [36] and that only a small number is highly linked, while the majority is only sparsely linked [35]. Taking the above into consideration, automating the process of searching for relevant datasets should be regarded as a crucial step in the development of the WoD that will facilitate Linked Data practitioners in initiating tasks such as Link Discovery.

In this spirit, methodologies [4,12,20,31] and online tools [7,8] that automatically recommend datasets for Link Discovery have been recently proposed. These use as source of evidence for determining datasets' relevancy, information such as datasets' instance/schema keywords [31], graph structure [12] and existing links between them Mountantonakis and Tzitzikas [27], Leme et al. [20].

However, a characteristic that carries rich semantics but remains unexploited is the geographic information available in datasets. According to Schmachtenberg et al. [36], W3C BasicGeo vocabulary,[5] which is one of the most common spatial ontologies, is used for georeferencing instances in more than 25% of LOD datasets. This paper explores the idea of leveraging the geographical information in datasets, specifically, the georeferenced location of instances expressed as point geometries, in order to recommend relevant WoD triplesets.

The impact of geographic information for deducing semantic relatedness at the instance or at the schema level has already identified in several contexts such as Link Discovery [13,43] and information retrieval [16,37]. According to the Tobler's first law of geography, "everything is related to everything else, but near things are more related than distant things" [40]. In Link Discovery, a common approach to determine the relatedness between instances is to calculate their geographic distance so as spatially near instances are more likely to represent the same thing [13]. Extending this for triplesets, we hypothesize that if a set contains instances that tend to be nearly located with the instances of another set (in other words, the two sets present similar spatial distribution) then the two triplesets are likely to contain semantically related instances. In the information retrieval scope, Ballatore et al. [2] define geo-semantic relatedness as: "Every term is geo-semantically related to all other terms, but terms that co-occur[6] with specifiable geographic relations are more related than other terms", where terms are lexemes that refer to specific entities or phenomena, such as rivers, accidents and buildings. Driven by Ballatore's definition, we formulate our hypothesis for triplesets containing instances that refer to specific entities or phenomena, i.e., for triplesets containing instances that are members of a specific conceptual category or class (rather than triplesets containing the sum of instances provided by a dataset, because a dataset contains instances that belong to diverse concepts or classes). Therefore, our hypothesis is formulated as:

> Pairs of classes whose instances present similar spatial distribution are more related than pairs of classes whose instances present dissimilar spatial distribution, in the sense that the former are more likely to contain semantically related instances

In this paper, we propose methods to support our hypothesis by computing a degree of geospatial relatedness between classes, so as the more similar the spatial distributions of two classes are, the more likely to contain semantically related instances, and thus the more relevant they are to be recommended for tasks such as link discovery.

Before formally expressing the problem, we clarify the terms that we use in this paper. According to the vocabulary of interlinked datasets (VoID),[7] a dataset is defined as a set of RDF triples published, maintained or aggregated by a single provider. Datasets are accessible through SPARQL Endpoints or RDF files and contain instances that, using the *rdf:type* predicate, are declared to be members of one or more classes. A class contains a subset of the dataset's instances that belong to the same conceptual category such as *Museums*, *Persons* or *Cities*. A spatial class is a class that contains spatial instances, that is, georeferenced instances whose locations are represented as vectors geometries (points, lines, or polygons), using spatial ontologies. Since the vast majority of instances in WoD are georeferenced as points, in this work, we deal only with spatial classes containing point geometries. For brevity, when we use the term *class* we will actually refer

[2] http://datahub.io/.

[3] http://lod-cloud.net.

[4] http://stats.lod2.eu.

[5] https://www.w3.org/2003/01/geo/.

[6] They refer in term co-occurrence in text corpuses.

[7] http://vocab.deri.ie/void#Dataset.

to the term *spatial class*. Formally, we define the problem as follows:

> Given a source class S and a set of target classes $T = \{T_1, T_2, \ldots, T_n\}$, rank target classes according to their degree of geospatial relatedness, so as the top ranked target classes are more probable to contain semantically related instances with S.

S can be any user-selected WoD spatial class for which one wants to get recommendations and T consists of (ideally) all spatial classes in the WoD.

The posed problem raises the following questions: (a) how to compute a geospatial relatedness score for classes? and (b) how this score can be computed efficiently so as to support recommendations among thousands of WoD classes? A naïve implementation to score classes according to their spatial distributions similarity, would require calculations on the exact locations of instances, i.e., pairwise calculations of Euclidean distances between instances locations, which is impractical for recommendations at web scale. Thus, we adopt an approach that first summarizes classes' spatial characteristics and then, based on these summaries, calculates a geospatial relatedness score between classes. We summarize two spatial characteristics of classes, namely their spatial extent and the spatial distribution of their instances locations. The spatial extent of a class is summarized by geometric approaches such as minimum bounding rectangles (MBR) and ConvexHulls. The spatial distribution of a class is summarized by a set of a QuadTree cell IDs that correspond to the cells containing class instances. The QuadTree covers the whole earth and consists of different-sized cells reflecting the spatial distribution of all WoD instances. The geospatial relatedness measures calculate the similarity of the classes' summaries, thus eliminating the need for calculating distances between the exact locations of instances. Specifically, we adapt and evaluate seven measures: number of common cells IDs, Jaccard Index, overlap coefficient, Poisson distribution probability, phi coefficient, pointwise mutual information, and mutual information. In the evaluation section, we show that the proposed approach provides high quality recommendations and that it generates ranked lists of relevant classes with approximately 35% higher mean average precision than simple baselines based on the textual and semantic similarity of class names. In addition, we propose a spatial filtering step that rules out "obviously" irrelevant (to the source class) target classes from subsequent calculations and from the ranked lists. Our experiments show that the proposed spatial filters reduce effectively the search space for relevant classes by 99%.

The rest of the paper is organized as follows: In Sect. 2, we present the related work in the dataset recommendation for link discovery domain and previous works on point sets similarity. In Sect. 3, we present our approach, which consists of methods that summarize spatial classes and measures that compute their geospatial relatedness. An implementation of our approach is presented in Sect. 4, including details on WoD "crawling" for the identification of available spatial classes, the construction of the QuadTree, class summarization and the recommendation algorithm. We evaluate our approach in Sect. 5, which includes an exploration of WoD spatial classes' characteristics; the formation of the ground truth; the baselines and the experiments setup; and the results of the experiments. In Sect. 6, we discuss the evaluation findings, and in Sect. 7, we indicate next steps and conclude.

2 Related Work

2.1 Dataset Recommendation for Link Discovery

This paper addresses the dataset recommendation for Link Discovery problem, which aims at the discovery of Web of Data datasets that may contain related instances so as to be suggested for the link discovery task. Typically, the input is a source dataset that is compared against a set of target datasets and the outcome is a (usually ranked) list of relevant (to the source dataset) target datasets. (In Link Discovery a source dataset is compared against a target dataset to return pairs of related instances.) We identify three main approaches in the existing literature, based on the "source of evidence" for determining dataset relevancy: (a) keyword-based (b) graph-based, and (c) linkage-based approaches.

Keyword-based approaches measure the textual similarity of instance or schema information between datasets. Nikolov and d'Aquin [31] and Nikolov et al. [32] identify an initial set of target datasets by issuing, to the Sig.ma semantic web index [41], keyword queries consisting of random labels extracted from the source dataset's entities. Then, they rank this initial set of target datasets by applying ontology matching techniques that assess the semantic similarity between classes (such as string similarity of labels and semantic relations defined in WordNet). Similarly to our work, they also recommend relevant classes among datasets. Ben Ellefi et al. [4] adopt dataset profiling techniques for characterizing datasets through a set of class labels and descriptions and they use these profiles to identify schema overlap between datasets. Initially, they identify a cluster of datasets that share schema classes with a given dataset by calculating a similarity measure based on term co-occurrence and WordNet semantic distance. Then, for each dataset in the identified cluster, they compute a dataset relevancy score based on cosine similarity of td*idf representations of dataset profiles to generate ranked lists of relevant datasets. As an additional contribution, they also return mappings between dataset's classes. A dataset recommendation tool, called DRX, which is also based on dataset profiles, was proposed by Caraballo et al. [8]. Other keyword-based approaches apply topic modelling methods [3,33] based on the assumption that similar datasets should have similar topics. For example, in TAPIOCA [33], a corpus of documents, where a document characterizes a dataset by its schema metadata (class and properties labels), is used as input to the latent dirichlet allocation (LDA) algorithm to create a topic model. The topic model preserves the distribution over topics for each dataset and the ranking order of the recommended datasets is determined by their topic distribution similarity. Keyword-based methodologies were also proposed by Mehdi et al. [26] and Martins et al. [25] and applied in the life sciences domain.

Graph-based approaches compare the similarity of datasets ontology graphs to determine whether two datasets are likely to contain related instances. Emaldi et al. [12] built on the assumption that "similar datasets should have a similar structure and include semantically similar resources and relationships" and exploit frequent subgraph mining techniques to find graph similarities among datasets. They extract frequent subgraphs from datasets and then they evaluate their similarity by computing the cost of transforming one graph to another. The lower the transformation cost, the higher the probability that two datasets are relevant.

Linkage-based approaches recommend relevant datasets by using as source of evidence existing links between datasets. Leme et al. [20] and Lopes et al. [23] build a Linked Data network, a graph where nodes represent datasets and edges denote the existence of links between them; the evidence about the existence of links between datasets is extracted from metadata in datahub. Based on this graph, Leme et al. [20] develop a measure based on Bayesian classifiers and Lopes et al. [23] adapts link prediction measures used in the Social Networks domain, such as the Jaccard and the Adamic–Adar coefficient, that rank target datasets according to the probability to be linked with the source dataset. The above works lay the ground for the development of TRTML, an online dataset recommendation tool [7]. Similarly to Lopes et al. [23], Liu et al. [22] construct a graph of linked datasets and apply link prediction measures. To increase the accuracy of the recommendations, they combine them with supervised classifiers, specifically bagging and Random Forests. In another work, Liu et al. [21] approach the problem from a recommender system's perspective; they construct a user–item matrix where both users and items are datasets and the rating values depend on the numbers of existing sameAs links between datasets, extracted by the Linked Open Data Cloud 2014 dump. Then, they predict the rating values, that is, the number of possible links, for new datasets by taking into account the corresponding ratings of similar datasets. The similarity of datasets is computed by the cosine distance on tf*idf-weighted vector model representations that contain datasets' schema terms (vocabularies, classes and properties). We note that dataset recommenders often combine more than one of the above described approaches [3,21,32]. For example, Nikolov et al. [32], additionally to keyword-based similarity measures, they also consider existing sameAs links between classes to determine their relevancy.

A fundamental difference of our approach to the aforementioned dataset recommendation for Link Discovery approaches is that we use as source of evidence for determining dataset relevancy the geographic information available in datasets. Since our approach is based on the topological similarity of classes, it can reveal relevant classes that cannot be identified by the other recommendation approaches, such as multilingual classes containing related instances (a limitation of the keyword-based approaches identified by Ben Ellefi et al. [4]) or topologically similar classes containing not only equivalent, but also not equivalently related instances. For example, the similarity of the spatial distributions of a *Universities* and a *CampusLibraries* class, or between an *Airports* and a *MeteorologicalStations* class (airports usually have in-premises meteorological stations) reveals their semantic relation. Indeed, a data provider might find it useful to interlink instances from the two classes by, say, a *affiliatedOrganization*[8] or a *partOf*[9] relation.

2.2 Point-Set Similarity

Several works propose summarizations of point-sets and metrics for computing their similarity but none of them focuses on the dataset recommendation for Link Discovery problem. Addressing Peer-to-Peer environments, Kufer and Henrich [18] combine geometric (e.g., minimum bounding rectangles) and space-partitioning (e.g., grids) summaries for point-sets, where a point-set is a collection of the georeferenced items of a peer. The proposed summaries are developed for queries like "Find peers that intersect with a query polygon" (range queries) or "Rank the top-k nearest peers to a query location" (kNN queries), which are irrelevant to the dataset recommendation for Link Discovery scenario because they do not measure the similarity of the spatial distributions of the sets. Zhu et al. [46] apply 27 spatial statistics on point-sets, where a point-set contains the locations of Gazetteers' feature types instances, in order to generate spatial semantic signatures for each feature type and evaluate their discriminative power for the identification of similar (or dissimilar) feature types so as to support ontology alignment. The proposed statistics are based on spatial point patterns (e.g., local intensity, Ripley's K), spatial autocorrelation (e.g., Moran's I, semivariograms) and spatial interactions

[8] University Ontology (https://www.cs.umd.edu/projects/plus/SHOE/onts/univ1.0.html).

[9] Dublin Core Metadata Initiative (http://www.dublincore.org/specifications/dublin-core/dcmi-terms/).

with other geographic feature types (e.g., count of distinct nearest feature types). However, as they state, these statistics are mostly descriptive and cannot be used alone (without combining them with feature type string and structural similarity approaches) for effective ontology alignment.

Sherif and Ngomo [38] compare various point-set distance measures, including Mean, Max, Average, Link and Hausdorff distance, for effective and efficient Link Discovery. In their work, a point-set represents the polygon geometry of an instance and the distance measures (which act on the exact locations of the points and not on point-sets summaries) are used to identify similar polygons in order to aid the Link Discovery process. To reduce the number of exact Hausdorff Distance calculations for large collections of point-sets, Adelfio et al. [1] propose the computation of an enhanced lower bound approximation (called E_{NH}LB) of the exact Hausdorff distance value, which acts on minimum bound rectangle (MBR)-based point-set summarizations. The above point-set distance functions (except from the mean distance) perform pairwise distance calculations between the exact location of the points included in the point-sets, which is inefficient for the dataset recommendation for Link Discovery scenario. Moreover, their effectiveness on recommending datasets for Link Discovery is not tested and have been proved to be ineffective in some contexts such as geo-social similarity [17].

Geo-social similarity refers to the problem of finding similar social network users and some works approach the problem by measuring the distance of point-sets, where a point-set consists of the locations of a user activities. Kanza et al. [17] propose and evaluate two novel distance measures for finding the k-most similar users to a given user: the mutually nearest distance and the QuadTree distance. Compared to our approach, their QuadTree distance requires the construction of a QuadTree for every user (in our case, we construct a single QuadTree, based on which we summarize the spatial distribution of class instances), which explodes the storage costs and the complexity of the similarity algorithm. Efstathiades et al. [11] introduces the spatio-textual point-set similarity join (STPSJoin) query, which seeks for users with similar sets of spatio-textual items (e.g., geolocated photos or tweets). The similarity between two users is calculated as the ratio of their matched items to the sum of their items. Two items match, if their spatial location is closer than a given distance and the textual similarity is above a given threshold. To reduce the comparisons only for items that satisfy the spatial condition they employ in their algorithms Grid and R-Trees indexes. In their experiments, they evaluate the execution time of the proposed algorithms and not the effectiveness in correctly identifying similar users.

None of the above described works on point-set similarity can be directly applied to the dataset recommendation for Link Discovery problem: Kufer and Henrich [18] target a different type of queries; Zhu et al. [46] do not provide enough discriminative power; Sherif and Ngomo [38] and Adelfio et al. [1] require distance calculations between the exact point locations; Kanza et al. [17] and Efstathiades et al. [11] involve increased storage or complexity costs. Here we present a novel point-sets similarity approach, where a point-set consists of the locations of a class instances, and we apply it in the dataset recommendation for Link Discovery problem. Our proposed approach can be useful in other problems that employ point-set similarity, such as geo-social similarity [17], dataset search [9], web tables extension [10,19], or data source selection for federated queries [14,34,42].

3 Geospatial Relatedness of Classes

As already noted, we compute a score of geospatial relatedness based on class summaries that capture spatial characteristics of classes, namely their spatial extent and the spatial distribution of their instances. In the following, we present the summarization methods for the spatial extent (Sect. 3.1) and the spatial distribution of classes (Sect. 3.2), and the summary-based measures of geospatial relatedness (Sect. 3.3).

3.1 Spatial Extent Summaries

The similarity of two classes' spatial distributions is affected by the existence of nearly located pairs of instances between the two classes. An efficient way to determine if two classes do not contain any pairs of nearly located instances is to examine their spatial extents. The spatial extent of a class is an area that encloses the location of all instances of the class. If the spatial extents of two classes are not overlapping, then these classes will not contain any pair of nearly located instances so their spatial distribution is dissimilar. For example, the classes *AirportsInFrance* and *AirportsInChina* are spatially extending in non-overlapping parts of the world, there is no possibility for them to contain nearly located instances, and thus they are not a good pair to recommend for Link Discovery.

Typically, the spatial extent of a point-set is represented by the minimum rectangle or the minimum polygon that encloses all the points of a set, i.e., the Minimum Bounding Rectangle (MBR) or the ConvexHull respectively (Fig. 1). MBR is a simpler geometry than ConvexHull, and thus it requires less storage and computation, but ConvexHull captures the spatial extent of a class more precisely. Compared to spatial distribution summaries (presented in Sect. 3.2), spatial extent summaries are much more "lightweight", and therefore they are suitable for the filtering phase of the recommendation algorithm (Sect. 4.4) for efficiently excluding non-related classes, i.e., classes with non-intersecting spa-

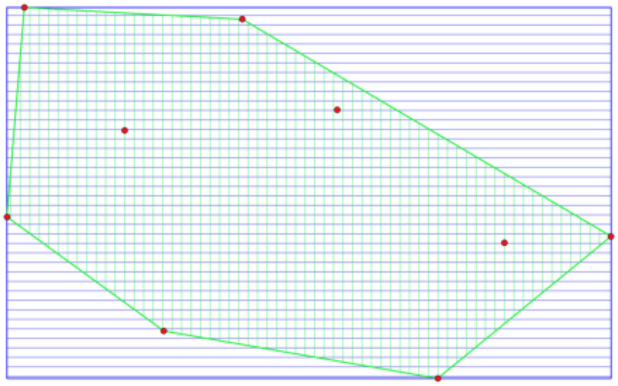

Fig. 1 Capturing the spatial extent of a class with MBR (horizontal lines) and ConvexHull (vertical lines). Points represent the locations of the class instances

tial extent. On the other side, spatial extent summaries are not "rich" enough for effectively calculating the similarity of two classes' spatial distribution.

3.2 Spatial Distribution Summaries

Next, we examine methods that summarize the spatial distribution of the class' instances by approximating their locations. Specifically, we employ space-partitioning techniques that discretize the earth's surface into disjoint cells where each cell is identified by a unique ID. Using the discretized view of the world as the underlying structure, we summarize each class spatial distribution by generating a set of cell IDs that correspond to the cells in which the class has instances.

A simple space partitioning technique, the Regular Grid, segments the space in equally sized cells. Since Web of Data instances can be located anywhere in the world, the grid should cover the whole earth. A Regular Grid with large-sized cells requires few cells to cover the earth's surface but each cell covers a large geographic area, which results in high information loss when approximating instances' locations. For example, a Regular Grid that consists of 10 km × 10 km cells, needs about five million cells to cover the earth's surface and each cell covers an area of 100 km^2 (approximately equal to the area of a big city like Barcelona), clearly a large area for approximating the location of an instance, such as a museum. A Regular Grid with small-sized cells approximates instance locations more precisely but needs more cells to cover earth surface which explodes the associated storage and the computational cost. For example, a 100 m × 100 m cell size Regular Grid where each cell covers an area of 0.01 km^2 (approximately equal to the area of a football stadium) is more suitable for approximating the location of a museum but it needs about 50 billion cells to cover earth.

At this point, we highlight two observations regarding instances' characteristics that should be taken into account for choosing among appropriate space partitioning structures. First, points represent zero-area instances but in reality instances correspond to objects that cover small or large areas. For example, a point instance about a museum in real word covers an area of few hundred m^2 and a point instance about a city covers an area of few (tens) of km^2 (even though representing large-area objects, such as cities, as points is not accurate, many data providers prefer to represent instances using points instead of polygons to gain in storage costs). After the examination of 100 randomly chosen WoD classes we roughly estimated the area size of the real world objects that their instances correspond: about 75% of classes contain instances that refer to "small-area" objects, covering areas ranging from some hundreds of m^2 to few km^2 (such as Churches, Airports and Neighborhoods), and about 25% to "large-area" objects, covering areas of tens to thousands km^2 (such as Cities, Mountains and Countries). Based on this observation, we deduce that a possible selection of space partitioning structures cell sizes for the precise and efficient summarization of instance locations should range from a few thousand m^2 to few thousand km^2. Large cells that cover areas of thousands km^2 cannot summarize instances precisely enough (leading to high information loss) while small cells that cover areas of few m^2 are too "pulverized" for approximating instance locations (leading to unnecessary explosion of storage and computational costs).

The second observation is that instances are not uniformly distributed in the earth's surface. Instances tend to concentrate in places where human activity is intense, such as city centers, and there are areas, such as oceans, where there are no instances at all. For high density areas, where many instances are located, small-sized cells would be a better choice for space discretization, while in low density areas, where few or no instances are located, space can be discretized using large-sized cells. For example, a cell that covers an area of 100 km^2 and overlaps with the city of London will contain too many instances from many different classes and therefore the summarization at the city of London will be too coarse. On the contrary, a cell that covers an area of 100 m^2 in the Pacific Ocean, provides no real value for class summarization and it just adds to storage and computational costs.

The latter observation leads us to examine another space-partitioning technique: the QuadTree index which segments space in not equally-sized cells. A QuadTree is an initially Regular Grid but each cell is split recursively in 4 sub-cells when a criterion is met. Using a QuadTree we can discretize earth surface unevenly, where high-density areas (high concentration of WoD instances) are covered by small-sized cells and low-density areas (low concentration of WoD instances) are covered by large-sized cells. To capture the spatial distribution of WoD instances, we specify as the cell split criterion the total number of WoD classes that contain instances in a cell. If a cell contains instances from many classes it means that the underlying area is dense, and the cell should be split so that the particular area to be covered by smaller cells. This process is executed recursively until a specified QuadTree depth (i.e., number of times that an initial cell can be split) is reached. The algorithm for the construction of the QuadTree is presented below. A possible QuadTree generation along with a discussion about parameter setting is described in Sect. 4.2.

Algorithm 1: QuadTree Construction

```
Input: A Regular Grid $R$ that consists of $v$ equally-sized cells,
       the set of the sample classes that will be used for the
       QuadTree construction $C$, the number of classes that
       triggers a cell split $n$, and the maximum QuadTree depth
       $D$
Output: The final QuadTree $Q$
begin
    /* Copy Regular grid $R$ to $Q$ */
    $Q = R$;
    /* Initialize a list $N$ that holds the
       number of classes that contain
       instances in each $Q$ cell where $n_i$
       refer to a $Q$ cell */
    $N = \{n_1 : 0, n_2 : 0, \ldots, n_v : 0\}$ ;
    /* Set current depth level */
    $d = 1$;
    while $d <= D$ do
        foreach $c_i \in C$ do
            Parse the location of $c_j$ instances;
            /* Add 1 to cells ($n_i$) if they
               contain instances of $c_j$ */
            $n_i = n_i + 1$;
        foreach $n_i \in N$ do
            if $n_i >= n$ then
                Split $n_i$ cell in 4 equally sized subcells;
        Replace the split cells with their new subcells in $Q$;
        /* Initialize $N$ including the new
           cells */
        $N = \{n_1 : 0, n_2 : 0, \ldots, n_v : 0\}$;
        /* Increase current depth level */
        $d = d + 1$;
    return $Q$;
```

After the creation of a space partitioning structure, say, a QuadTree, we summarize the spatial distribution of a class by generating a set of cell IDs that correspond to the QuadTree cells that contain class instances. Figure 2 presents the summarization process for a fictional class *S*, containing 12 georeferenced instances (Fig. 2b), on a hypothetical global QuadTree index consisting of 25 cells (Fig. 2a). By overlaying the class instances on the QuadTree we generate the class's summary, that is, the set of the QuadTree cell IDs that contain class' instances (Fig. 2c). We follow the same procedure (using the same global QuadTree index) to summarize all WoD classes' spatial distribution.

In the next section, we propose measures that, based on these summaries, calculate the geospatial relatedness of classes. Thus, we turn the problem of comparing the spatial distribution of classes into comparing sets of cell IDs, which correspond to summaries of classes' spatial distribution.

3.3 Geospatial Relatedness Measures

To compare the similarity of the spatial distributions of two classes, we focus on their common geographical area, because in areas not covered by both classes, according to our methodology, we cannot compute similarity. So, for example, the comparison of the *BanksInEurope* and *BanksInGreece* classes' spatial distribution is performed in the intersecting area of their spatial extent: that is, Greece. The intersection of the spatial extent of two classes is computed as the intersection polygon (I) of classes' ConvexHulls and the comparison of the classes' spatial distribution summaries is restricted to the QuadTree cells overlapping with I. Figure 3 illustrates an example of comparing a *Squares* class with a *Triangles* class. Let S and T represent the spatial distribution summary sets of the two classes respectively, that include only the cells overlapping with the I polygon; $|S|$ and $|T|$ corresponds to their summaries set sizes in the I polygon, respectively. The set of cells that are common in the S and T summaries is represented as C and the size of the set of common cells as $|C|$. The set of the QuadTree cells overlapping with I is represented as Q and its size as $|Q|$. Notice that the number of two classes summaries common cells is always less or equal to the size of the smallest class summary ($|C| \leq \text{Min}(|S|, |T|)$) and the size of the S and T summaries is always smaller or equal to the number of the QuadTree cells ($|S|, |T| \leq |Q|$).

3.3.1 The Simple Approach

The simplest way to measure the geospatial relatedness between two classes is to count the number of cells that their summaries have in common. The intuition behind this measure is that the more the common cells (thus, pairs of nearly located instances), the more similar their spatial distribution. The set of common cells C is the intersection of the S and T summaries and its size is:

$$|C| = |S \cap T| \tag{1}$$

$|C|$ does not take into account the size of summaries, which is an important factor for assessing the geospatial relatedness between classes. As a general rule, large summaries are more likely to have more common cells than small summaries. For example, two summaries that contain thousands of cells (e.g., a class containing the location of all museums in the world

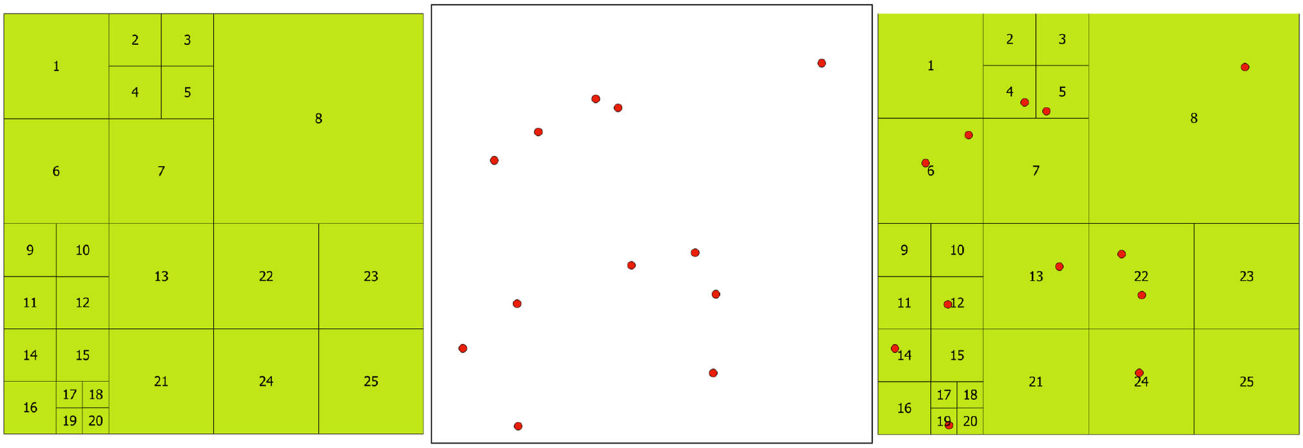

Fig. 2 Summarizing the spatial distribution of a class. **a** A hypothetical Global QuadTree with the respective Cell IDs. **b** The location of the fictional S class 12 instances. **c** overlaying instances on the Quad Tree. The summary for the S class is represented by the set $S = \{4, 5, 6, 8, 12, 13, 14, 19, 22, 24\}$ with size of $|S| = 10$. We note that if a cell contains more than one instances is maintained once in the summary set

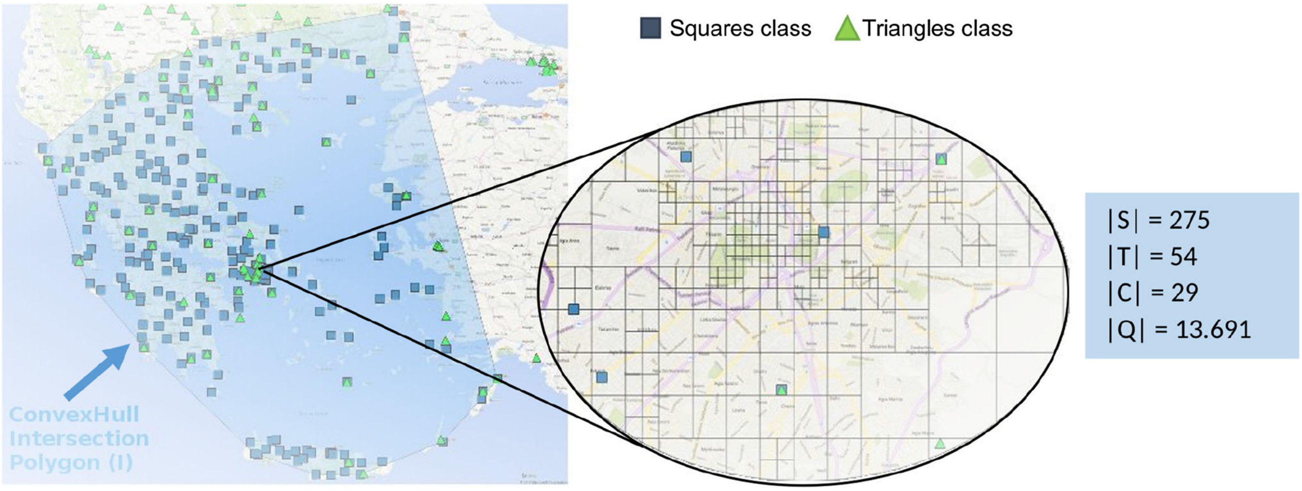

Fig. 3 Comparison of the spatial distribution of two fictional squares and triangles classes. The comparison is performed in their ConvexHulls intersection polygon (I). $|S|$ is the number of S summary cells (cells that contain square instances) in I polygon. $|T|$ is the number of T summary cells (cells that contain triangle instances) in I polygon. $|C|$ is the number of S and T common summary cells (cells that contain both squares and triangle instances). $|Q|$ is the number of QuadTree cells included in I polygon

and a class containing the location of all banks in the world) may have more common cells than two summaries that contain just a few instances (e.g., an airports in Greece and a major airports in Greece class). Next, we propose measures taking into account the sizes of the summaries of the compared classes.

3.3.2 The Set Similarity Approach

Since classes are represented by summaries, that is, sets of cell IDs, we can compute a geospatial relatedness score between classes by applying common set similarity measures on their summaries. We can assume that the more similar these summaries sets are, the more related the respective classes are. A well-known set similarity measure, Jaccard Index (J), is defined as the size of the intersection of two sets divided by their union. We compute Jaccard Index as the intersection of two summaries (common cells) divided by their union:

$$J(S, T) = \frac{|S \cap T|}{|S \cup T|} = \frac{|C|}{|S| + |T| - |C|} \tag{2}$$

Jaccard Index returns decimal values from 0 to 1: 0 means that two summaries have no common cells and 1 that they have all their cells common. A drawback of the Jaccard Index is that it does not perform well when the sizes of two summaries are significantly different; so, it fails in revealing relevant classes when the first is a subset of the other. For example, a class about *CapitalCitiesAirports* will contain several sameAs instances with a class about *WorldAirports*; however, their J value is low because the intersection of their summaries is small compared to their union. A more appropriate set similarity measure for comparing different-sized sets is the overlap coefficient (O), which is defined as the intersection of two sets divided by the size of the smaller set. We compute O as the number of the intersection of two summaries (common cells) divided by the size of the smaller summary:

$$O(S,T) = \frac{|S \cap T|}{\mathrm{Min}(|S|,|T|)} = \frac{|C|}{\mathrm{Min}(|S|,|T|)} \tag{3}$$

Overlap Coefficient returns decimal values from 0 to 1; 0 means that the summaries are completely disjoint and 1 that a summary is a perfect subset of the other.

The presented measures so far, do not take into account the frequency of instances in a geographic area, which is a useful parameter for the calculation of a geospatial relatedness score between classes. Neglecting this parameter can result in misleading conclusions about the similarity of the spatial distributions and thus about the semantic relevance of two classes, for instance, when calculating the overlap coefficient for dense classes. Illustrating this with an example, a class about *TrafficLightsInMadrid* probably contain instances in every QuadTree cell that overlaps with Madrid. Thus, comparing any class with the *TrafficLightsInMadrid* class will return high O value (because the compared class is a subset of the *TrafficLightsInMadrid* class); In this case, high O value is not an indication that the classes contain related instances. In the following, we include the frequency of class' instances in a given area parameter to calculate the geospatial relatedness score between classes.

3.3.3 The Probability Approach

The frequency of class instances in a geographic area is calculated as the number of the class summary cells divided by the total number of the QuadTree cells in this area. Using the example of Fig. 3, the frequency of the "*Squares*" class in the spatial area covered by I is:

$$f_S = \frac{|S|}{|Q|} \tag{4}$$

The frequency of the *Triangles* class for the same area (f_T) is estimated accordingly by substituting in Eq. (4), $|S|$ by $|T|$. For any class (e.g., the *Squares* class), there are two possible outcomes: Either a cell contains class instances or does not contain class instances. Event S_1 consists of all the outcomes (cells) where the class has instances and is equal to the class summary S; event S_0 consists of the remaining outcomes (cells in which the class has no instances) and is equal to the set S'. In any given geographical area covered by $|Q|$ cells, events S_1 and S_0, and their respective sets, are complementary, so $|S| + |S'| = |Q|$. Based on the assumption that the probability of each of the two possible outcomes for a given cell is equal, the probability of the S_1 event is:

$$P(X = S_1) = f_S = \frac{|S|}{|Q|} \tag{5}$$

The probability of the S_0 event is the complement of S_1 so:

$$\begin{aligned} P(X = S_0) &= 1 - P(X = S_1) = 1 - \frac{|S|}{|Q|} \\ &= \frac{|Q| - |S|}{|Q|} = \frac{|S'|}{|Q|} \end{aligned} \tag{6}$$

Now that we have turned class frequencies into probabilities of events, we can use the probability property that states that two events, A and B, are independent when the occurrence of A is not affected by the occurrence of B and, if two events are independent, the product of their probabilities is equal to their joint probability:

$$P(A)P(B) = P(A \cap B) \tag{7}$$

Let us suppose that we examine the relation between two events: event S_1 that an S class has instances in cells and event T_1 that a T class has instances in cells in a given area. If the events S_1 and T_1 are independent, the occurrence of the S_1 cells is not affected by the occurrence of T_1 cells, so the location of the instances of the S class is not affected by the location of the instances of the T class, and thus we can deduce that the two classes are not related. Else, if the events S_1 and T_1 are not independent, the occurrence of S_1 cells affects the occurrence of T_1 cells, so the location of the instances of the S class is affected by the location of the instances of the T class and we can assume that the classes are to some degree related. We can also assume that the more dependent event S_1 and T_1, the more related the classes S and T are. The independence event formula (Eq. 7) will help us determine the degree of dependency between the event S_1 and T_1, and therefore the degree of geospatial relatedness between the two classes. We do this by comparing the difference between the number of the common cells ($|C|$) that two classes, S and T, actually have, with the number of the common cells that the two classes should have if the events S_1 and T_1 were independent (Ci). We can suppose that the bigger the difference between $|C|$ and Ci, the more dependent events S_1 and T_1 are. Since we already known $|C|$, we only need to calculate Ci based on the Eq. (7):

$$P(S_1 \cap T_1) = P(S_1)P(T_1) \Leftrightarrow \frac{Ci}{|Q|} = \frac{|S|}{|Q|}\frac{|T|}{|Q|} \Leftrightarrow Ci = \frac{|S||T|}{|Q|} \tag{8}$$

A straightforward way to compare $|C|$ and Ci is to compute their ratio R (Eq. 9). R is a positive decimal indicating how much bigger $|C|$ is from Ci. When R is close to 1, the number of actual common cells is almost equal the number of common cells in case the events S_1 and T_1 were independent, so the instances' location of one class is not affected by the instances' location of the other class. The bigger the R is, the bigger the difference between $|C|$ and Ci, so the more dependent events S_1 and T_1 are, and thus the instances' location of one class is affected by the instances' location of the other class.

$$R = \frac{|C|}{Ci} \tag{9}$$

Equation (10) shows that the logarithm of R is equal to the pointwise mutual information (PMI) of the events S_1 and T_1, $\text{PMI}_{(S1,T1)}$. PMI is a measure of association between events calculated as the logarithm of the probability of the actual co-occurrence of two events, $P(S_1, T_1)$, divided by the product of their probabilities [6]. PMI results to zero when two events are independent and large absolute PMI values imply strong (negative or positive) correlation between events.

$$\begin{aligned}\text{PMI}(S_1, T_1) &= \log\frac{P(S_1, T_1)}{P(S_1)P(T_1)} = \log\frac{\frac{|C|}{|Q|}}{\frac{|S|}{|Q|}\frac{|T|}{|Q|}} \\ &= \log\frac{|C|}{\frac{|S||T|}{|Q|}} = \log\frac{|C|}{Ci} = log R\end{aligned} \tag{10}$$

Another way to quantify the difference between $|C|$ and Ci is to estimate the probability for two summaries, S and T, to actually have $|C|$ or more common cells knowing that if they were independent, they should have Ci common cells. A well-known probability distribution, the Poisson, calculates the probability of a number of events to occur in a space where events are occurring at a constant and known rate λ. Notice that Ci value also represents the expected (or the mean) number of common cells between two randomly generated classes (with the same summary size with S and T respectively) in a given area. Then, the desired probability follows a Poisson distribution where the random variable X is the number of the common cells that two summaries have in a given spatial area and $\lambda = Ci$ represents the expected number of common cells for two summaries in that area. The probability of two summaries, S and T, to have C or more common cells, is the inverse cumulative Poisson probability $P(X \geq |C|)$ (Eq. 11) calculated as the sum of the individual Poisson probabilities $P(X = x)$ where x takes values from $|C|$ to $\text{Min}(|S|, |T|)$ (we remind that $|C|$ is always less or equal to $\text{Min}(|S|, |T|)$). High inverse cumulative probability means that the actual number of common cells is likely to be produced by two random classes and that $|C|$ is close to the expected number of common cells (C_i). On the contrary, a low probability means that having $|C|$ or more common cells is unlikely to be the product of two random classes, $|C|$ differs significantly from the expected number of common cells (C_i) and thus the two summaries, and consequently the respective classes, are geospatially related.

Table 1 The 2 × 2 contingency table for the X and Y variables

	Y_1	Y_0	Total
X_1	n_{11}	n_{10}	$n_{11} + n_{10}$
X_0	n_{01}	n_{00}	$n_{01} + n_{00}$
Total	$n_{11} + n_{01}$	$n_{10} + n_{00}$	$n_{11} + n_{01} + n_{10} + n_{00}$

$$P(X \geq x) = \sum_{x=|C|}^{\text{Min}(|S|,|T|)} P(x) \quad \text{where} \quad P(x) = \frac{e^{-Ci}Ci^x}{x!} \tag{11}$$

3.3.4 The Association of Binary Variables Approach

A class summary can be also represented by a binary vector variable corresponding to cells where value 1 denotes that the class has instances in a cell and 0 denotes that the class has not instances in a cell. Then, the problem of estimating the geospatial relatedness between two classes, can be approached by calculating their respective binary variables association for the cells (observations) that intersect in a given geographical area. If two binary variables present strong positive correlation, the values of the first variable are affected by the values of the second variable (that is, the location of the instances of one class affects positively the location of the instances of the other class), so the respective classes are geospatially related.

A standard metric for estimating the association between two binary variables is the phi coefficient (Φ) [39]. Φ is calculated by constructing the contingency table for the two variables X and Y, which displays the frequency counts (n) for all outcome variable combinations (Table 1) such as n_{11} is the number of observations where both X and Y are 1, n_{10} is the number of observations where X is 1 and Y is 0 and so on, and then by applying the Φ formula (Eq. 12). Phi coefficient values ranges from − 1 to 1, where − 1 indicates strong negative correlation, 0 no correlation and 1 strong positive correlation between two variables.

Table 2 The 2 × 2 contingency table for the S and T variables

	T_1	T_0	Total
S_1	$\|C\|$	$\|S\| - \|C\|$	$\|S\|$
S_0	$\|T\| - \|C\|$	$\|Q\| - \|S\| - \|T\| + \|C\|$	$\|Q\| - \|S\| = \|S'\|$
Total	$\|T\|$	$\|Q\| - \|T\| = \|T'\|$	$\|Q\|$

$$\Phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{(n_{11}+n_{10})(n_{01}+n_{00})(n_{11}+n_{01})(n_{10}+n_{00})}} \tag{12}$$

To compute the association between the binary variables for the S and T classes we construct the contingency table (Table 2) where S_1 represent the cells for which the S variable takes the value 1 (S class has instances in the cell), S_0 represent the cells for which the S variable takes the value 0 (S class has no instances in the cell) and T_1 and T_0 represent the corresponding values for the T variable. The contingency table is filled using the already known values: the number of cells that both S and T are 1 is the number of their common cells ($|C|$), the number of cells that S is 1, and T is 0 is the size of the S summary minus the number of the common cells ($|S| - |C|$), the total number that S is 1 is the size of the S summary ($|S|$) and so on. The total number of observations (cells) equals to the total number of QuadTree cells that are included in the given spatial area ($|Q|$), so S_0 is equal to the total number of QuadTree cells in the given area minus the cells where S is 1 ($|Q| - |S| = |S'|$) and accordingly T_0 is equal to $|Q| - |T| = |T'|$. Equation (13) presents the formula for calculating Φ for the S and T variables after substituting the variables of Eq. (12) with the corresponding values of Table 2.

$$\begin{aligned}\Phi &= \frac{|C|(|Q| - |S| - |T| + |C|) - (|S| - |C|)(|T| - |C|)}{\sqrt{|S|(|Q| - |S|)|T|(|Q| - |T|)}}\\ &= \frac{|C| * |Q| - |S| * |T|}{\sqrt{|S||S'||T||T'|}}\end{aligned} \tag{13}$$

The second binary variable association metric is the mutual information (MI). Previously, we calculated the association for one combination of S and T variable outcomes, namely the $\mathrm{PMI}(S_1, T_1)$ where S_1 corresponds to the event that the S class contains instances in a cell and T_1 corresponds to the event that the T class contains instances in a cell. Equation (14) presents the formula for calculating the association between all possible S and T event combination outcomes, that is, the MI of S and T variables, calculated as the weighted sum of the PMI for all possible event combinations namely: $\mathrm{PMI}(S_1, T_1)$, $\mathrm{PMI}(S_1, T_0)$, $\mathrm{PMI}(S_0, T_1)$ and $\mathrm{PMI}(S_0, T_0)$. Individual event probabilities correspond to values in the marginal cells of Table 2 divided by the total number of cells (e.g., $P(S_0) = |S'|/|Q|$) and joint event probabilities to the values in the inner cells of the contingency table divided by the total number of cells (e.g., $P(S_1, T_0) = |S| - |C|/|Q|$). MI returns non negative decimals, where values close to zero indicate that the variables are independent and high values that variables, are correlated, and thus classes are geospatially related.

$$\mathrm{MI} = \sum_{x=0}^{1}\sum_{y=0}^{1} P(Sx, Ty)\mathrm{PMI}(Sx, Ty)$$
$$\text{where} \quad \mathrm{PMI}(Sx, Ty) = \log \frac{P(Sx, Ty)}{P(Sx)P(Ty)} \tag{14}$$

To sum up, in this section we proposed seven measures, which exploit class spatial summaries to calculate the degree of geospatial relatedness between classes. These measures are the: (a) number of common cells (Eq. 1), (b) Jaccard Index (Eq. 2), (c) overlap coefficient (Eq. 3), (d) pointwise mutual information (Eq. 10), (e) Poisson distribution probability (Eq. 11), (f) Phi coefficient (Eq. 13), and (g) mutual information (Eq. 14). In Sect. 5, we evaluate each measure effectiveness for accurately recommending relevant classes for link discovery.

4 Class Recommender Implementation

In this section, we present the implementation of a class recommender for link discovery,[10] which consists of the following components:

1. The *Spatial Class Locator* component identifies and catalogs available spatial classes in the Web of Data (Sect. 4.1).
2. The *QuadTree Construction* component generates a QuadTree upon which the summarization of classes spatial distribution is based (Sect. 4.2) and
3. The *Spatial Class Summarization* component creates summaries for the catalogued classes (Sect. 4.3).
4. The *Recommendation Algorithm* component first filters non-geospatially related classes and then recommends a ranked list of classes for Link Discovery to a source class (Sect. 4.4).

We note that the first three components are executed offline and only the *Recommendation Algorithm* is executed

[10] http://geo-aegean.aegean.gr:8080/WoDSpatialClassRecommender/.

at runtime on user request (even though pre-generated lists of relevant classes for the already summarized classes can be created offline). However, if the input (source) class to the *Recommendation Algorithm* is not selected from the list of catalogued and summarized classes,[11] the *Spatial Class Summarization* component is also executed at runtime to construct summaries for the not catalogued source class.

4.1 Spatial Class Locator

The goal of the *Spatial Class Locator* component is to identify as many as possible (if possible, all) WoD spatial classes in order to form a large pool of target classes, that is, classes which are candidates for recommendations. A rich source of information about existing WoD datasets is data catalogs such as datahub.io,[12] a CKAN-based online catalog that preserves datasets' metadata including title, description, and links to their data sources (e.g., SPARQL Endpoint URLs). Datahub.io content is exposed though CKAN API, which is exploited by the component to initially locate WoD datasets provided through SPARQL Endpoints. Then, *Spatial Class Locator* issues a SPARQL query to each located dataset to retrieve the list of the spatial classes (if any) that the dataset contains. The query returns classes that contain spatial instances that have been geo-referenced by reusing one of the most common spatial ontologies, listed in LOV[13] and LOV4IoT,[14] and namely are the W3C Basic Geo, NeoGeo,[15] GeoSPARQL,[16] OrdnanceSurvey,[17] GeoNames[18] and GeoRSS.[19] The following SPARQL query returns a list of a dataset's classes that contain georeferenced instances using the W3C BasicGeo ontology:[20]

```
SELECT DISTINCT ?class
WHERE {
   ?s <http://www.w3.org/2003/01/geo/wgs84_pos#long> ?x.
   ?s <http://www.w3.org/2003/01/geo/wgs84_pos#lat> ?y.
   ?s a ?class
}
```

Spatial Class Locator initially located 684 datasets provided through SPARQL endpoints. More than 50% of them are not stable, no longer available or could not be parsed so were not further considered. From the remaining datasets, *Spatial Class Locator* identified 54 ones containing georefernced instances in totally 20,640 spatial classes[21] (Table 3). This number does not include classes that contain less than five instances (because they contain a small number of instances and therefore are not valuable recommendations for link discovery) and classes that contain more than 100,000 instances (because these classes are few and usually correspond to top-level classes such as owl:thing). Table 4 shows the number of identified spatial classes per spatial ontology used for instance georeference. We note that an instance location may be georeferenced by more than one spatial ontology. For example, in DBPedia the location of instances is represented by either W3C Basic Geo, GeoRSS or both spatial ontologies. Classes that contain instances that use two spatial ontologies are preserved twice, each containing the instances of a particular spatial ontology because the set of instances of a class that use the W3C Basic Geo ontology may be different from the set of instances of the same class that use the GeoRSS ontology. Specifically, of the 11,692 distinct identified spatial classes, 8948 of them preserved twice but only 167 of them contain exactly the same instances using different ontologies.

4.2 QuadTree Construction

The *QuadTree Construction* component is executed once to generate the QuadTree upon which the summarization of classes' spatial distribution will be based. In Sect. 3.2, we presented the algorithm for the construction of a QuadTree without specifying any values for the required parameters. In this section, we discuss the parameters that we set in order to construct the QuadTree for the specific implementation. The first parameter is the *size of the cells of the initial Regular Grid*. Notice that this size is also the size of the final QuadTree not split cells, that is, cells that cover extremely low-density areas where few or no instances are located. Since we want to cover these areas by large sized cells (in order to keep storage and computational costs low), we set the initial Regular Grid cell size equal to an area of 2500 km^2 (50 × 50 km edge size), which is approximately equal to the size of a small country, such as Luxemburg. The next parameter is the *maximum QuadTree depth*, that is, the maximum times an initial cell is allowed to split. Notice that the *size of the cells of the initial Regular Grid* and the *maximum QuadTree depth* specify the possible sizes of the QuadTree cells. Since high-density areas, such as city centers, should be covered by small cells that cover areas of few hundred m^2 (approximately the size of a soccer stadium), we set the maximum

[11] The implementation can be extended so as the source class to be any point spatial dataset specified by the user (such as a personal shapefile, a geoJSON file, a Web Feature Service or a spatial class from a non-identified SPARQL endpoint).

[12] https://old.datahub.io/.

[13] http://lov.okfn.org.

[14] http://lov4iot.appspot.com/?p=ontologies.

[15] http://geovocab.org/.

[16] http://www.geosparql.org/.

[17] http://data.ordnancesurvey.co.uk/ontology.

[18] http://www.geonames.org/ontology/.

[19] http://www.georss.org/rdf_rss1.html.

[20] The respective SPARQL queries for the rest ontologies are available at https://github.com/vkopsachilis/WoDSpatialClassRecommender.

[21] The full list of identified spatial classes is available at https://github.com/vkopsachilis/WoDSpatialClassRecommender.

Table 3 WoD datasets and the number of the spatial classes that each dataset contains

Dataset	Spatial classes	Dataset	Spatial classes
DBpedia	15,488	Social semantic web thesaurus	10
LinkLion	898	Lotico	10
DBpedia Wikidata	544	LOD for tourists in Castilla y Leon	9
URIBurner	523	EEA vocabularies	7
DBpedia in Greek	504	Environment agency bathing water quality	7
LinkedGeoData	349	ASCDC_LOD	6
DBpedia in French	305	Datos.bcn.cl	6
DBpedia in Dutch	206	OxPoints (University of Oxford)	6
Serendipity	180	linkedarc.net archaeological datasets	6
Open Data of Ecuador	180	Events calendar for the University Oxford	6
Universidad Técnica Particular de Loja	180	ISPRA—The Italian Data Buoy Network (RON)	5
DBpedia in Portuguese	176	Isidore	4
DBpedia in Spanish	176	Indicators Academic Process	4
DBpedia in Japanese	172	OpenMobileNetwork	3
Alexandria Digital Library Gazetteer	143	DBpedia Commons	3
GeoLinkedData	117	Hellenic Fire Brigade	3
houses of culture Caceres	67	World War 1 as Linked Open Data	3
Perfil del Contratante Cáceres	67	Alpine Ski Racers of Austria	2
Influence Tracker Dataset	54	Enipedia—Energy Industry Data	2
transport.data.gov.uk	37	CRTM	2
education.data.gov.uk	36	Courts thesaurus	2
Linked Logainm	31	Hellenic Police	1
Shoah victims' names	28	Imagesnippets Image Descriptions	1
Dutch Ships and Sailors	22	Linked Open Aalto Data Service	1
DBpedia in Basque	18	Geological Survey of Austria Thesaurus	1
Verrijkt Koninkrijk	14	Linked Crowdsourced Data	1
EIONET RDF Data	13	AEMET metereological dataset	1
Total number of spatial classes			20,640

Table 4 Number of spatial classes per spatial ontology

Spatial ontology	Class frequency
W3C Basic Geo	11,316
GeoRSS	9249
Geonames	43
NeoGeo	31
GeoSPARQL	1
Total number of spatial classes	20,640

Table 5 QuadTree cell sizes and their frequencies

SN	Edge size (km)	Area (km^2)	Frequency
1	50.00	2.500	233,301
2	25.00	625	59,922
3	12.50	156.25	102,347
4	6.25	39.06	197,988
5	3.12	9.76	262,953
6	1.56	2.43	304,635
7	0.78	0.61	329,864
8	0.39	0.15	315,120
9	0.20	0.04	250,880
	Total cells		2,057,010

QuadTree depth to 9, which results in the QuadTree cell sizes listed in Table 5. The smallest-sized cells covers an area of 0.04 km^2 (200 × 200 m edge size).

The *sample of classes* parameter of the QuadTree construction algorithm should be as large as possible in order to precisely simulate the distribution of the WoD instances (high- and low-density areas). Since QuadTree creation is a single-run offline process, we set this parameter as the maximum possible, that is, the 20,640 identified spatial classes by the *Spatial Class Locator* component. The algorithm retrieves the list of longitude and latitude coordinates for each class' instances by issuing a SPARQL query to the associated SPARQL endpoint. For example, the following query returns the coordinates of the instances of the "Fjord" class (provided by the LinkedGeoData dataset) georefenced using the W3C Basic Geo ontology.[22]

```
SELECT ?x ?y
WHERE {
   ?s <http://www.w3.org/2003/01/geo/wgs84_pos#long> ?x.
   ?s <http://www.w3.org/2003/01/geo/wgs84_pos#lat> ?y.
   ?s a < http://linkedgeodata.org/ontology/Fjord>
}
```

The last parameter is the cell split criterion, that is, the *maximum number of classes* allowed to have instances in a cell before triggering a split. Setting the maximum allowed number of classes too low will result in many splits, consecutively in many and small QuadTree cells, and thus in increased storage and computational costs. Setting this parameter too high will result in few splits, consecutively in few and large QuadTree cells, and thus in reduced summarization precision. Knowing that a cell can contain instances from a maximum of 20,640 classes, for this implementation we set the cell split criterion value arbitrarily to 30 classes.

Running the QuadTree construction algorithm with the above parameters, resulted in a QuadTree that covers earth with approximately 2 million cells (recall that a 10 km × 10 km cell Regular Grid covers earth with approximately 5 million cells). Table 5 lists the QuadTree cells sizes along with their respective frequencies, that is, number of cells per size. We notice that a significant proportion of the QuadTree cells are the not split cells that cover large and low-density areas and that the frequency for the rest cell sizes is smoothly increasing as the cell size decreases until reaching the highest frequency which refer to cells covering an area of 0.61 km^2. A snapshot of the constructed QuadTree is depicted in Fig. 4.

4.3 Spatial Class Summarization

The *Spatial Class Summarization* component creates summaries for the identified spatial classes based on the location of their instances. First, it retrieves the list of longitude and latitude coordinates for each class' instances (as described in Sect. 4.2) and then, based on these lists of coordinates, it constructs and stores summaries for each class that capture (a) class' spatial extent by generating its ConvexHull and (b) class' spatial distribution by imposing these coordinates on the QuadTree, according to the process described in Sect. 3.2.

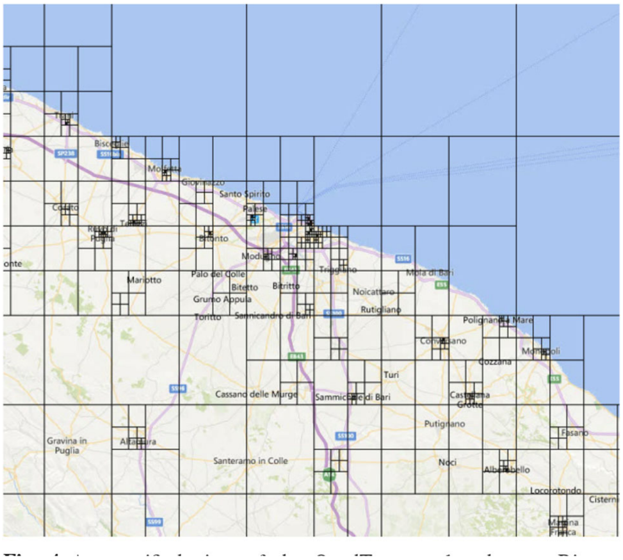

Fig. 4 A magnified view of the QuadTree overlayed on a Bing BaseMap in the area of Puglia, Italy. Low-density areas are covered by large cells (e.g., sea) and high-density areas are covered by smaller cells (e.g., cities)

4.4 Recommendation Algorithm

The *Recommendation Algorithm* compares the summaries of a source class, S, with the summaries of a set of target classes, $T = \{T_1, T_2, \ldots, T_n\}$, to recommend relevant classes for Link Discovery, ranked by their degree of geospatial relatedness. The source class S can be selected from the list of 20,640 identified and summarized spatial classes. The set of target classes consists of the rest identified WoD spatial classes, so its size is $|T| = 20{,}639$ classes. However, T size is reduced after excluding classes that are provided by the same dataset as the S class, because we assume that the user wants to get recommendation for classes that exist in different datasets from the source class. We note that one could add more preferences to the recommendation algorithm, such as to not recommend classes that are already contain interlinked instances or classes that are linked with, say, a parent/child relation, but suggesting an exhaustive list of such preferences is not the focus of this paper. The *Recommendation Algorithm* is executed in two phases: the filtering and the ranking phase.

4.4.1 Filtering Classes

The goal of the filtering phase is to rule out "obviously" irrelevant (to the source class) target classes so as to reduce the search space and the size of the ranked recommendation lists. The algorithm implements the following two spatial filters.

Recall that two classes are not relevant if their spatial extents are disjoint, so the first spatial filter states that:

[22] The respective SPARQL queries for the rest ontologies are available at https://github.com/vkopsachilis/WoDSpatialClassRecommender.

A target class is removed from T, if it's ConvexHull does not intersect with the ConvexHull of the source class

For the implementation of the first spatial filter, we only utilize the spatial extent class summaries to exclude all target classes that their ConvexHulls present a null intersection polygon with the ConvexHull of the source class. However, even if two classes' ConvexHulls intersect, it may be the case that all the instances of the source class are located far from all the instances of the target class. Since there are no nearly located instances between the two classes, we can deduce that the pair is not good candidate to be recommended. To assess whether two classes contain pairs of nearly-located instances, we utilize the spatial distribution summaries, and particularly the number of common cells. Thus, the general form of the second spatial filter is:

A target class is removed from T, if its summary has less than a minimum number of common cells with the summary of the source class.

The *minimum number of common cells* is an algorithm parameter that can be set depending on the user preferences. A high value removes many target classes from T and results in smaller lists of recommended classes with fewer irrelevant classes but increased probability of missed relevant classes. A low value results in larger lists of recommended classes with more irrelevant classes but reduced probability of missed relevant classes. For this implementation, we set the minimum number of common cells parameter to two, so classes with no or just one common cells are removed. In Sect. 5, we experiment and discuss the impact of this parameter setting.

4.4.2 Ranking Classes

In the ranking phase, the algorithm computes a geospatial relatedness score between the source class and each of the remaining target classes and ranks them accordingly. The score is computed using any of the metrics (defined by the user) presented in Sect. 3.3. To compute the geospatial relatedness score between two classes, S and T_i, the algorithm calculates: (a) the polygon (I), that is, the intersection polygon of the source and target class ConvexHulls, (b) the number of the common cells (C) of the source and target class summaries, (c) the number of cells of the source class summary S contained in I, (d) the number of cells of the target class summary $|T_i|$ contained in I, (e) the number of the QuadTree cells $|Q|$ contained in I. The execution flow of the *Recommendation Algorithm* is presented below.

Algorithm 2: Recommendation Algorithm

```
Input: A source class $S$, a set of target classes
       $T = T_1, T_2, ...T_n$, the minimum number of common
       cells $m$ of the second spatial filter, and the metric $M$
       (one of C, J, O, PD, Phi, PMI, MI)for the calculation
       of the geospatial relatedness score between two classes
Output: A ranked list of relevant T classes to S, $L$
begin
    /* Filtering Phase                                   */
    foreach $T_i \in T$ do
        Calculate the intersection polygon $I_i$ of the S and $T_i$
        ConvexHulls ;
        if $I_i == null$ then
            Remove $T_i$ from $T$ ;
    foreach $T_i \in T$ do
        Calculate the number of common cells $C_i$ of S and $T_i$ ;
        if $C_i < m$ then
            Remove $T_i$ from $T$ ;
    /* Ranking Phase                                     */
    foreach $T_i \in T$ do
        Calculate the number of S summary cells in $I_i$ (|S|) ;
        Calculate the number of $T_i$ summary cells in $I_i$ ($|T_i|$) ;
        Calculate the number of QuadTree cells in $I_i$ ($|Q_i|$) ;
    Calculate the geospatial relatedness score $M_i$ by applying
    the given metric $M$ for S and $T_i$ and add it to $L$ ;
    Rank $L$ based on $M_i$ ;
    return $L$;
```

5 Evaluation

5.1 Spatial Classes Characteristics

In this section, we provide insights about the characteristics of the 20,640 identified and summarized spatial classes that will be helpful for the formation of the ground truth and the discussion of the experiment results. First, we examine the distribution of spatial classes according to their size, that is, the number of instances they contain. We classify classes in 7 bins, ranging from small (classes containing less than 50 instances) to large (classes containing more than 50,000 instances). Figure 5a presents the defined bins and the corresponding class frequencies and shows that about 78% of the classes contain less than 200 instances while only 1.31% contain more than 10,000 instances. Second, we examine the distribution of the classes according to their summaries size, that is, the number of QuadTree cells that each class summary contains. We classify classes using the same bins as previously, ranging from small class summaries (containing less than 50 cells) to large class summaries (containing more than 50,000 cells). Figure 5b presents the frequencies of classes according to their summary size and shows that about 81% of class summaries contain less than 200 cells and less than 1% of class summaries contain more than 10,000 cells. Notice that the class summary size distribution resembles the class size distribution. A class summary size (number of summary cells) is equal to the class size (number of instances) if all class instances are located in different cells and smaller if some of the class instances are located in the same cells. Figure 5c presents the distribution of classes according to the rounded mean number of instances per cell (calculated as the number of a class instances divided by the number of its summary cells and rounded to the closest integer) and shows that the vast majority of classes (about 86%) contain approximately one instance per QuadTree cell.

Last, we examine the size of classes' spatial extent, that is, their convexHulls area in km^2. Figure 5d presents the distribution of classes' spatial extent classified in 5 bins each representing an area roughly equal to a common geographical notion, ranging from small areas, covering medium sized cities, to large areas, covering the whole world. Most classes are "global" (about 34%), many classes cover areas approximately equal to continents, counties and regions (about 20% each) and only a few classes are "local" (about 2%), covering small areas such as cities. We conclude, by mentioning one more observation that we take into account in order to form our ground truth: Class size (Fig. 5a) and spatial extent distributions (Fig. 5d) are not strongly correlated (Pearson Coefficient resulted in 0.30), meaning that the size of a class does not strongly affect its spatial extent size and vice versa; thus, in the WoD we meet a variety of spatial classes regarding their characteristics such as classes with few instances covering the whole earth, classes with many instances covering small geographical areas and so on.

5.2 Ground Truth

A usual approach to form a gold standard for evaluating dataset recommenders for Link Discovery performance is to extract metadata about existing links provided by data catalogs, such as the LOD [3,4,12] or datahub [8,20,23]. Nevertheless, this approach presents the false positives problem; i.e., many pairs of datasets that actually contain related instances are not included in the (usually manually created) metadata. To overcome this problem, we formed our ground truth by thoroughly examining a sample of the identified classes. In order to reduce the bias of the sample, we took into consideration the spatial class characteristics presented in Sect. 5.1; specifically, we selected randomly 20 classes (out of 20,640) that belong to various: (a) datasets, (b) class size categories and (c) spatial extent categories, proportionally to the frequencies listed in Table 3, Fig. 5a, d, respectively. The selected classes are used in the experiments as source classes, that is, classes for which we aim to get recommendations and listed in Table 6 along with the dataset they belong, their number of instances, summary size and spatial extent area. We note that we excluded classes provided by (English) DBpedia (they represent about 75% of the total identified classes) in order to avoid forming a sample containing many classes from a single dataset.

For the annotation of pairs of relevant classes, we had to examine each one of the 20 classes with the rest of the identified spatial classes that would require $20 \times 20{,}639 = 412,870$ manual examinations. However, after applying the spatial filters presented in Sect. 4.4 this number is significantly reduced to 2441. We note that the filtering step may have removed relevant classes from the sample. However, our results are not affected by this, since we don't focus on the recall of the methodology (that is, the effectiveness in recommending all related classes in the Web of Data), but on the effectiveness in recommending precise rankings of relevant classes. Furthermore, an exhaustive manual examination of related classes in WoD would be infeasible. The third column of Table 7 shows the number of remaining classes after filtering for each source class, that is, the number of manual examinations for each source class.

To examine whether a pair of classes is relevant (contain related instances) or not relevant (does not contain related instances), we manually inspected their contents. Specifically, we retrieved the instance set of each class (their labels and point locations) and we search for related instances: two instances are related if they refer to the same object (e.g., the same museum) or to semantically close objects (e.g., a university and its campus library). The judgment about instances relatedness was based on their semantic characteristics, that is, their labels and class names, and was aided by the geographic context (by projecting the instances on a Google basemap using the QGIS software). In this way, we ensured that instances with the same label are not considered to be related if they refer to different real-world objects (e.g., two different cities with the same name). We also annotate as relevant a pair of classes, if it contains semantically related instances with dissimilar spatial distribution (due to wrong or vague georeference, or not containing inherently stationary instances, such as persons). Nevertheless, this was hardly the case in our sample, because of the filtering phase. The last column of Table 7 presents the number of the annotated relevant classes for each source class. As an example of the manual examination outcome, we provide the list of relevant classes to the Fjord class (Table 8).[23]

5.3 Baselines

To evaluate the performance of the proposed spatial-based measures we set up three simple non-spatial baselines. The first generates lists of classes randomly ranked. Random ranking provides an indication of how much the proposed measures improve the average precision of the ranking in

[23] The full ground truth list is available at https://github.com/vkopsachilis/WoDSpatialClassRecommender.

Table 6 Sample classes selected for ground truth

SN	Source class	Dataset	Number of instances	Summary size	Spatial extent (km^2)
1	http://www.wikidata.org/entity/Q44613	DBpedia in Dutch	275	111	28,934,571
2	http://dbpedia.org/class/yago/PopulatedPlacesInTheLesbosPrefecture	DBpedia in Greek	15	15	12,079
3	http://schema.org/Library	DBpedia in Japanese	589	524	48,487,372
4	http://dbpedia.org/ontology/Locomotive	DBpedia Wikidata	57	46	36,445,749
5	http://dbpedia.org/ontology/Organ	DBpedia Wikidata	113	113	9,054,483
6	http://education.data.gov.uk/def/school/IndependentSchoolType_Music	education.data.gov.uk	37	36	298,169
7	http://greek-lod.math.auth.gr/fire-brigade/resource/subdivisions	Hellenic Fire Brigade	281	275	605,417
8	http://linkedgeodata.org/ontology/ChurchHall	LinkedGeoData	49	49	2,013,850
9	http://linkedgeodata.org/ontology/TricycleStation	LinkedGeoData	250	134	328,304
10	http://linkedgeodata.org/ontology/Fjord	LinkLion	111	84	601,967
11	http://linkedgeodata.org/ontology/Newsstand	LinkLion	77	58	18,922,362
12	http://linkedgeodata.org/ontology/MineralSpring	LinkLion	46	10	6494
13	http://linkedgeodata.org/ontology/GrouseButt	LinkLion	47	5	455
14	http://ns.ox.ac.uk/namespace/oxpoints/2009/02/owl#Room	OxPoints (University of Oxford)	75	28	37
15	http://opendata.caceres.es/def/ontomunicipio#ActividadDeportiva	Perfil del Contratante Cáceres	126	35	57
16	http://schema.org/PlaceOfWorship	Perfil del Contratante Cáceres	59	48	983
17	http://schema.org/TouristAttraction	Perfil del Contratante Cáceres	144	53	151
18	http://schema.org/Festival	Serendipity	18	13	75,169,606
19	http://linkedgeodata.org/ontology/City	Shoah victims’ names	94	94	1,109,817
20	http://transport.data.gov.uk/def/naptan/FerryTerminalDockEntrance	transport.data.gov.uk	104	91	1,130,248

Table 7 Ground truth analysis

SN	Source class	Number of classes after filtering	Number of relevant classes
1	http://www.wikidata.org/entity/Q44613	147	78
2	http://dbpedia.org/class/yago/PopulatedPlacesInTheLesbosPrefecture	18	7
3	http://schema.org/Library	214	35
4	http://dbpedia.org/ontology/Locomotive	43	14
5	http://dbpedia.org/ontology/Organ	256	44
6	http://education.data.gov.uk/def/school/IndependentSchoolType_Music	148	52
7	http://greek-lod.math.auth.gr/fire-brigade/resource/subdivisions	296	2
8	http://linkedgeodata.org/ontology/ChurchHall	90	1
9	http://linkedgeodata.org/ontology/TricycleStation	175	4
10	http://linkedgeodata.org/ontology/Fjord	89	8
11	http://linkedgeodata.org/ontology/Newsstand	82	1
12	http://linkedgeodata.org/ontology/MineralSpring	6	2
13	http://linkedgeodata.org/ontology/GrouseButt	10	0
14	http://ns.ox.ac.uk/namespace/oxpoints/2009/02/owl#Room	175	25
15	http://opendata.caceres.es/def/ontomunicipio#ActividadDeportiva	48	6
16	http://schema.org/PlaceOfWorship	75	7
17	http://schema.org/TouristAttraction	76	20
18	http://schema.org/Festival	84	84
19	http://linkedgeodata.org/ontology/City	293	35
20	http://transport.data.gov.uk/def/naptan/FerryTerminalDockEntrance	116	11
	Total	2441	436

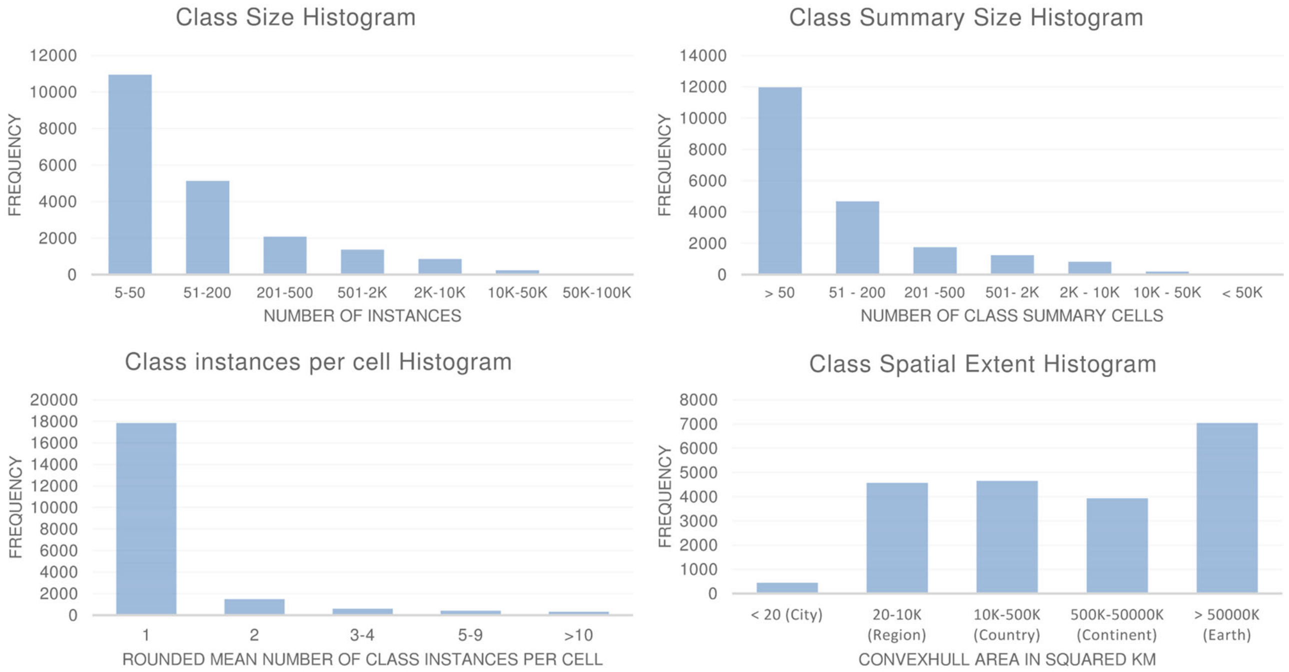

Fig. 5 **a** Histogram of classes based on number of instances. **b** Histogram of classes based on summary sizes. **c** Histogram of classes based on the rounded mean number of class instances per cell. **d** Histogram of classes spatial extent

Table 8 Relevant classes for the "http://linkedgeodata.org/ontology/Fjord" class provided from the "Linklion" Dataset

Class	Dataset
http://adl-gazetteer.geog.ucsb.edu/ontology/fjord	Alexandria Digital Library (ADL) Gazetteer
http://adl-gazetteer.geog.ucsb.edu/ontology/bar_physiographic_	Alexandria Digital Library (ADL) Gazetteer
http://adl-gazetteer.geog.ucsb.edu/ontology/physiographic_feature	Alexandria Digital Library (ADL) Gazetteer
http://dbpedia.org/class/yago/Fjord109281104 (W3C Basic Geo)	DBPedia
http://dbpedia.org/class/yago/Fjord109281104 (GeoRSS)	DBPedia
http://dbpedia.org/class/yago/Inlet109313716 (W3C Basic Geo)	DBPedia
http://dbpedia.org/class/yago/Inlet109313716 (GeoRSS)	DBPedia
http://linkedgeodata.org/ontology/NaturalThing	LinkedGeoData

the specific experiment setting than no ranking at all. The next two baselines, normalized Levensthein distance and WordNet-based similarity, generate ranked lists based on the textual and semantic similarity of class names respectively and provide an indication about the performance of the spatial measures compared with simple non-spatial measures. The input for these two baselines is the classes' names, which are extracted by parsing the last part of their URI, that is, the part after the last "/" or "#", and further edited by applying basic string cleansing, including removal of numbers and special characters and conversion to lower case. For the calculation of the WordNet-based Similarity between two class names, we additionally applied word segmentation and stemming using the PorterStemmer. An example of the class name extraction procedure for two class URIs is presented in Table 9.

Levensthein distance (LD) is a common string similarity metric that calculates the number of transformations required to transform one string to another. We normalize Levensthein Distance (NLD) by dividing the LD of two strings, s_1 and s_2 (the cleaned class names) with the length of the larger string.

$$\mathrm{NLD}(s_1, s_2) = \frac{\mathrm{LD}(s_1, s_2)}{\mathrm{MaxLength}(s_1, s_2)} \tag{15}$$

To calculate the WordNet-based similarity score of two words we use the WuPalmer similarity [45] implemented in the WS4J library.[24] If class names are phrases, that is, they include more than one word (e.g., "popul plac in the lesbo prefecture"), we calculate their similarity by dividing the sum

[24] https://github.com/emir-munoz/ws4j.

of pairwise word comparisons with the product of the two phrases size (number of words). If C_1 represent the word set of a class and C_2 represent the word set of a second class, their WordNet-based similarity (WS) is calculated as:

$$\mathrm{WS} = \frac{\sum_{i=1}^{|C_1|}\sum_{j=1}^{|C_2|}\mathrm{WuPalmer}(C_{1i}, C_{2j})}{|C_1||C_2|} \tag{16}$$

5.4 Experimental Setup

We evaluate the effectiveness of the proposed geospatial approach in identifying semantically related classes and in generating lists of recommended classes to a source class, where the relevant classes are ranked higher than the irrelevant classes. The experiment is setup as follows: We run the *Recommendation Algorithm* for each of the 20 source classes selected in the ground truth. In the filtering phase, we maintain the default filters (Sect. 4.4). On Sect. 5.5.1, we evaluate the reduction of the search space of the filtering step and on Sect. 5.6 we examine the effect of the minimum number of common cells parameter of the second spatial filter.

For each source class, we rank the remaining target classes 10 times: each, for the 7 proposed geo-semantic relatedness measures (common cells (C), Jaccard Index (J), overlap coefficient (O), Poisson distribution (PD), Phi coefficient (Φ), pointwise mutual information (PMI) and mutual information (MI)) and for 3 baselines (random ranking (Rnd), normalized Levensthein distance (NLD), and wordnet-based similarity (WS)). We, then, evaluate each measure performance by calculating the mean average precision (MAP) [24] of its resulted rankings for the 20 source classes. MAP is a single value indicator of a measure's effectiveness in producing rankings where relevant classes are ranked higher than irrelevant classes and returns scores between 0 and 1 where 0 indicates a poor ranking and 1 a perfect ranking. The MAP of each measure is the average of all source classes average precisions (AP) where AP is the average of the precision values at each relevant class position in a ranking and is expressed as follows:

$$\mathrm{MAP} = \frac{\sum_{s=0}^{N}\mathrm{AP}(s)}{N} \quad \text{where } \mathrm{AP}(s) = \frac{\sum_{k=1}^{n}(P(k)\mathrm{Rel}(k))}{Sr} \tag{17}$$

s refers to a source class, N is the number of the total source classes (in this experiment 20), Sr is the number of total relevant classes to a source class, k represents a position in the ranking list, $P(k)$ the precision at k position and $Rel(k)$ is a flag equal to 1 when the class at k position is relevant and 0 otherwise.

Table 9 Class URIs cleansing examples

	Class 1	Class 2
Class URI	http://dbpedia.org/class/yago/PopulatedPlacesInTheLesbosPrefecture	http://dbpedia.org/class/yago/WikicatEducationalInstitutionsEstablishedIn1960
Class name	PopulatedPlacesInTheLesbosPrefecture	WikicatEducationalInstitutionsEstablishedIn1960
Cleaned class name	populatedplacesinthelesbosprefecture	wikicateducationalinstitutionsestablishedin
Word segmentation	populated places in the lesbos prefecture	wikicat educational institutions established in
Stemming	popul plac in the lesbo prefecture	wikicat educ institute establish in

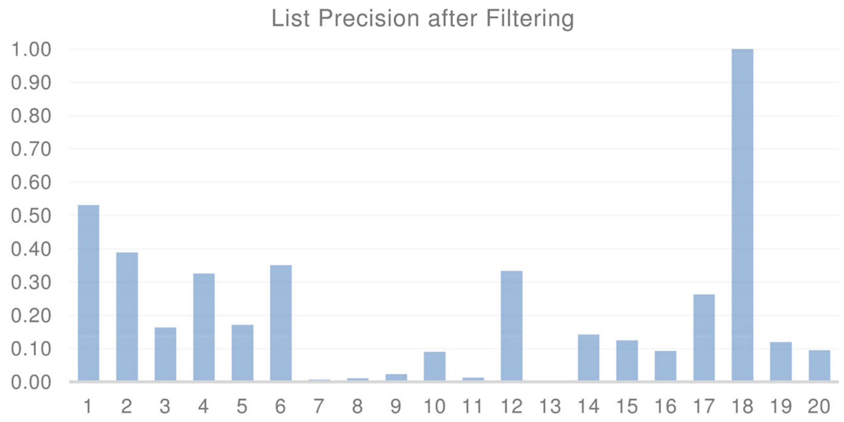

Fig. 6 Recommendation lists precisions for each source class. The numbers on the horizontal axis refer to the serial number (SN) of the source classes listed in Table 7

5.5 Results

5.5.1 Filtering Phase

In the filtering phase, the *Recommendation Algorithm* applies spatial filters to remove non-geospatially related classes from subsequent calculations, reducing thus the search space for relevant classes and the size of the recommendation lists. Using the default filter settings, it removes, on average for each source class, 20,516 classes (99.4% of the total number of identified WoD classes) and thus it reduces the search space (and the size of the recommendation lists), on average for each source class, to 123 classes (0.6% of the initial pool of target classes). According to the ground truth (Table 7), on average for each source class, 21% of the remaining classes after filtering were annotated as relevant classes; therefore, the precision of each recommendation list is, on average for each source class, 0.21. Figure 6 presents the precision of each source class recommendation list, calculated as the number of relevant classes divided by the number of the remaining target classes after filtering. The precision of the recommendation list for the *GrouseButt* class (number 13) precision is 0 since no relevant classes found in the ground truth. The precision for the *Festival* class (number 18) is 1, that is, all remaining classes after filtering are relevant, which means that all classes that their spatial extent intersects and their summaries have two or more common cells with the *Festival* class, contain related instances. This can be explained by taking into account the characteristics of the *Festival* class, presented in Table 6. *Festival* is a very small class regarding its size (it contains 18 instances) and a very large class regarding its spatial extend (its instances are distributed all over the world), and therefore a coincidence of their instances in two or more common cells is an indication that the classes are related.

5.5.2 Ranking Phase

The goal of the ranking phase is to rank the remaining (after the filtering phase) classes, so as relevant classes are positioned higher than the irrelevant classes. The ranking order is determined by the relatedness scores between the source and each of the target classes calculated by one of the proposed measures. Table 10 presents the average precisions (AP) of the ranked lists for each source class and for each of the seven spatial measures and the three baselines. The last row shows the Mean average precision (MAP) value for each measure. The bold underlined scores indicate the highest values for each source class and measure. If, for a given measure, the score of two target classes is the same, their ranking order is resolved by their scores on the other measures with the following order: (a) PD, (b) PMI, (c) Phi, and (d) Rnd for a spatial measure and (a) NLD, (b) WS and (c) Rnd for a baseline measure.

Table 10 shows that AP is the same regardless the applied measure for two source classes: the Festival class where all classes in the ranked list are relevant (so all metrics achieve AP equal to 1) and the GrousButt class where there are no relevant classes (so all metrics achieve AP equal to 0). Random ranking MAP equals to 0.30 that means that the most effective measure, PD (Poisson Distribution Probability) with 0.62 MAP, improve rankings more than 100% compared to not ranking at all. Also, PD outperforms the more effective baseline, NLD (Normalized Levensthein Distance) with 0.46 MAP, approximately by 35%. All spatial measures are more effective for ranking relevant classes compared to the simple textual or semantic baselines, which is an indication about the effectiveness of the spatial approach compared to non-spatial approaches.

Table 10 shows that all spatial measures perform worse than random ranking for one class, NewsStand, while for two classes, FerryTerminalDockEntrance and MineralSpring, some spatial measures perform equal or worse to random ranking. Normalized Levensthein Distance, and Wordnet-Based Similarity perform better than spatial mea-

Table 10 Average precision for each source class and measure and mean average precision for each measure

SN	Class	Spatial measures							Baselines (with spatial filtering)		
		C	J	O	Phi	PD	PMI	MI	Rnd	NLD	WS
1	Q44613	0.88	0.89	0.71	0.82	**0.92**	0.88	0.74	0.57	0.62	0.59
2	PopulatedPlacesInTheLesbosPrefecture	0.74	0.98	**0.93**	**0.93**	**0.93**	**0.93**	0.93	0.58	0.25	0.28
3	Library	0.54	0.69	0.63	0.66	**0.78**	0.74	0.62	0.19	0.54	0.55
4	Locomotive	0.79	**0.96**	0.79	0.95	0.89	0.95	0.89	0.33	0.38	0.38
5	Organ	**0.48**	0.34	0.29	0.33	0.42	**0.48**	0.32	0.23	0.12	0.13
6	IndependentSchoolType_Music	0.62	0.86	0.68	0.82	**0.88**	**0.88**	0.72	0.42	0.63	0.29
7	subdivisions	0.06	0.11	**1.00**	0.31	0.58	**1.00**	0.09	0.06	0.03	0.03
8	ChurchHall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.02	1.00	1.00
9	TricycleStation	0.30	0.30	0.30	0.29	0.30	0.11	0.30	0.04	**0.57**	0.27
10	Fjord	0.57	0.61	0.35	0.60	**0.67**	0.23	**0.67**	0.09	0.53	**0.67**
11	Newsstand	0.03	0.03	0.03	0.03	0.03	0.04	0.03	0.50	**1.00**	0.02
12	MineralSpring	**1.00**	**1.00**	0.83	0.83	**1.00**	0.83	**1.00**	0.83	0.83	0.67
13	GrouseButt	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	Room	0.37	0.39	0.56	**0.73**	0.68	0.53	0.57	0.16	0.23	0.29
15	ActividadDeportiva	0.52	0.78	0.49	0.76	**0.83**	0.71	0.69	0.27	0.62	0.56
16	PlaceOfWorship	0.59	0.48	**0.67**	0.57	0.59	0.55	0.57	0.32	0.24	0.24
17	TouristAttraction	0.38	0.32	0.66	0.47	0.59	**0.72**	0.45	0.26	0.27	0.35
18	Festival	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
19	City	0.15	0.25	0.19	0.25	0.22	0.23	0.25	0.12	0.23	**0.41**
20	FerryTerminalDockEntrance	0.09	0.13	0.07	0.12	0.16	**0.19**	0.11	0.11	0.10	0.13
MAP (Mean average precision)		0.51	0.56	0.56	0.59	**0.62**	0.60	0.55	0.30	0.46	0.39

sures for three classes: TricycleStation, Newsstand and City. Spatial measures underperform for classes that contain large real world objects, for instance, the City class, because the summarization of instances on the QuadTree fails to index related instances in the same cells. In such cases, different providers may choose far locations for georeferencing the same instance. For example, in one dataset the point of a city may correspond to the location of the town hall and on another dataset to the city's polygon centroid. Spatial measures perform better than baselines for 13 source classes and are more effective when relevant classes have different class names (for example, many relevant classes to the Locomotive class have names such as TramStop, Railing and Mean of Transportation, which achieve low scores in the baseline metrics). Table 10 reveals that the top 3 performing ranking measures are PD (Poisson Distribution Probability), PMI (pointwise mutual information) and Phi (Phi coefficient). These measures are also the more stable, since they achieve very low average precision (below 0.3) fewer times compared to the others measures: PD achieves very low AP for 3 classes, PMI for 5 and Φ for 4. Also, PD and PMI achieve higher AP the most times, for 7 and 6 source classes respectively. An example of the recommendation algorithm output is presented in Table 11 that includes the Top 20 (of the total 89) recommended classes to the "Fjord" class ranked according to the PD (Poisson distribution probability) score. Note that the last relevant class (out of 8 total) ranked at the 15th place.

Next, we examine the average precision of each measure at various recall points (Fig. 7) by interpolating the average precision at 11 recall points (0, 0.1, 0.2, …, 1) [24] for each source class and calculating the average of each recall point for the 20 source classes. Figure 7 shows that PD and PMI are the most effective measures at all recall points. It also shows, that all spatial measures achieve higher average precisions at all recall points compared to the NLD and WS baselines, and to the random ranking, which performs much worse at all recall points.

We note that the above MAP values for the baseline measures are affected by the number of the remaining classes after the spatial filtering step. In a slightly different experiment setting, we calculate the MAP (mean average precision) of the baselines, based on the same ground truth, without applying spatial filters, thus the ranked lists of recommended classes are comprised of all WoD classes minus the classes from the same dataset with the source class. In this setting, the MAP for random ranking, NLD and WS reduced to 0.002, 0.08 and 0.06 respectively because it was estimated on lists comprised of approximately 20,640 classes for each source class (instead of on average 123 classes after applying spatial filtering). This finding strengthens our argument that spatial

Table 11 Top 20 recommended classes for the Fjord class, ranked based on PD metric values

Rank	Class	Dataset	PD
1	**http://linkedgeodata.org/ontology/NaturalThing**	**LinkedGeoData**	**0**
2	**http://adl-gazetteer.geog.ucsb.edu/ontology/fjord**	**ADL Gazetteer**	**0**
3	**http://adl-gazetteer.geog.ucsb.edu/ontology/physiographic_feature**	**ADL Gazetteer**	**0**
4	http://adl-gazetteer.geog.ucsb.edu/ontology/valley	ADL Gazetteer	0
5	http://adl-gazetteer.geog.ucsb.edu/ontology/cape	ADL Gazetteer	0
6	http://linkedgeodata.org/ontology/Farm	LinkedGeoData	5.55E−16
7	http://adl-gazetteer.geog.ucsb.edu/ontology/mountain_summit	ADL Gazetteer	2.85E−12
8	http://adl-gazetteer.geog.ucsb.edu/ontology/reef	ADL Gazetteer	7.18E−09
9	http://linkedgeodata.org/ontology/Ruins	LinkedGeoData	1.36E−08
10	**http://dbpedia.org/class/yago/Fjord109281104(W3CBasicGeo)**	**DBpedia**	**5.29E−07**
11	**http://dbpedia.org/class/yago/Inlet109313716(W3CBasicGeo)**	**DBpedia**	**5.29E−07**
12	http://adl-gazetteer.geog.ucsb.edu/ontology/cliff	ADL Gazetteer	5.61E−07
13	**http://dbpedia.org/class/yago/Inlet109313716 (GeoRSS)**	**DBpedia**	**9.40E−07**
14	**http://dbpedia.org/class/yago/Fjord109281104 (GeoRSS)**	**DBpedia**	**9.40E−07**
15	**http://adl-gazetteer.geog.ucsb.edu/ontology/bar_physiographic_**	**ADL Gazetteer**	**1.66E−06**
16	http://adl-gazetteer.geog.ucsb.edu/ontology/gap	ADL Gazetteer	2.16E−06
17	http://www.geonames.org/ontology#V	LinkedGeoData	3.72E−06
18	http://dbpedia.org/class/yago/Bay109215664 (W3C Basic Geo)	DBpedia	1.09E−05
19	http://dbpedia.org/class/yago/Bay109215664 (GeoRSS)	DBpedia	1.09E−05
20	http://adl-gazetteer.geog.ucsb.edu/ontology/shrubland	ADL Gazetteer	1.18E−05

Classes in bold are the ground truth relevant classes

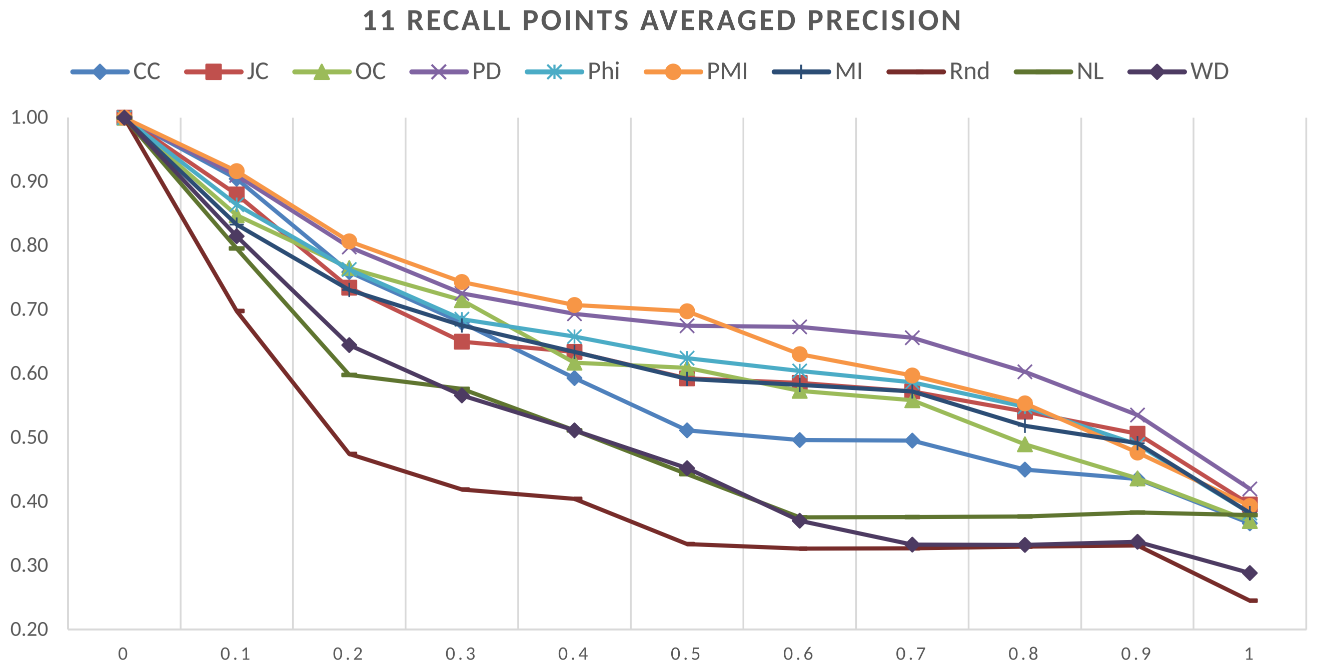

Fig. 7 Eleven recall points interpolated average precision for all measures. The horizantal axis contains the 11 recall point and the vertical axis the corresponding precision levels

information can be effectively used for recommending relevant WoD classes for Link Discovery; more importantly, highlights the importance of spatial filtering that significantly reduces the search space for relevant classes, while at the same time ensures that the subsequent ranking phase generate high precision recommendation lists.

5.6 Spatial Filtering Parameter

In Sect. 5.5.1, we presented the search space reduction, the size of the recommendation lists and the recommendation lists precision when the minimum number of common cells parameter of the second spatial filter is set to two. Table 12 summarizes the respective values for different *minimum number of common cells* settings. Setting lower parameter values (for example, removing classes with less than one common cells, that is, classes that do not have any common cells), results in bigger recommendation lists where more irrelevant classes are included (lower precision) but with decreased probability of removing relevant classes. On the contrary, providing higher parameter values (for example, removing classes with less than three common cells), results in smaller recommendation lists where less irrelevant classes are included (highest precision) but with increased probability of removing some relevant classes.

6 Discussion

To the best of our knowledge, this work is the first that exploits spatial information to recommend Web of Data classes for link discovery. We built on the hypothesis that pairs of classes with similar spatial distribution are more related than pairs of classes with dissimilar spatial distribution, in the sense that the former are more likely to contain semantically related instances. To support our hypothesis, we presented methods that summarize spatial classes and measure the degree of geospatial relatedness between classes. Recommended classes for Link Discovery are ranked according their degree of geospatial relatedness with the source class. Our evaluation results validate our hypothesis that geospatially related classes contain semantically related instances, and indicate that the proposed methods can provide high quality recommendations. Specifically, geospatial relatedness measures achieve mean average precision up to 62%, when simple baselines based on the textual and semantic similarity of class names achieve MAP up to 46% for ranking a set of classes formed after applying spatial filtering. A strong feature of the geospatial approach, which is not possible without exploiting the geospatial information in datasets, is the spatial filtering that reduces effectively and efficiently the search space for relevant classes (about 99%) by removing classes with not overlapping spatial extent and with few pairs of nearly located instances.

Each dataset recommendation approach, discussed in Sect. 2.1, can capture different aspects of datasets relatedness. A contribution of the geospatial approach is that it examines the topological relatedness of classes, which is ignored by the other approaches; so, it can reveal relevant classes described in different languages or classes that contain instances related not only with sameAs links. Some examples from our experiments include the reccomendation of a *Universities* and a *Libraries* class, an *Organ* and a *Church* class, and a *TouristAtrraction* and a *Park* class. The proposed geospatial approach is less effective than other approaches when instances are vaguely or erroneous georeferenced; for example, when classes contain large area real-world objects, such as cities, which are represented as points, the spatial approach is less effective in identifying their spatial distribution similarity.

In this work, we focused on presenting the building blocks of an effective spatially enabled recommendation process, leaving for future work aspects such as performance optimization and experiments about the associated storage and complexity cost. This paragraph sketches some issues that should be taken into account. First, the main computational burden of our approach refers to the generation of the QuadTree and the creation of the summaries for all WoD classes. Roughly estimated, the execution of the QuadTree construction algorithm and the class summarization component for the 20,640 classes, in a single AMD 64-bit Windows of 6GB RAM machine, requires about a week each. On the positive side, both are executed once and offline, so there are no runtime overheads. Runtime costs are associated with the execution of the *Recommendation Algorithm*, which compares a class with the rest WoD classes. However, runtime costs are reduced because of the spatial filtering phase, which rules out a large number of irrelevant classes; thus, subsequent calculations, that is, the calculation of geospatial relatedness score between classes, are performed for only a small subset of the total WoD classes. Moreover, geospatial relatedness score calculations are performed on class summary sets (that is, sets of cell IDs), which is much more efficient than calculating class similarity based on distances (e.g., Euclidean) between the exact locations of class instances. The execution time of the *Recommendation Algorithm* depends on the remaining after filtering classes' number and sizes, and, roughly estimated, on the above-described machine, varies from few seconds to some minutes.

For the calculation of a geospatial relatedness score between classes, we proposed seven measures that compare class summary sets, including the common cells (C), Jaccard Index (J), overlap coefficient (O), Poisson distribution probability (PD), pointwise mutual information (PMI), mutual information (MI), and Phi coefficient (ϕ). Overall, the

Table 12 Mean number of recommended classes, search space reduction and recommendation lists precision (on average) for different *minimum number of common cells* parameter values

Minimum number of common cells	Mean number of recommended classes	Search space reduction (%)	Recommendation lists precision
1	290	98.6	0.16
2	123	99.4	0.21
3	78	99.6	0.27
4	56	99.7	0.33

most effective measures are the "probability-based" PD and PMI . However, each measure generates different rankings for each source class and additionally, for different sources classes, different measures perform better (for example, PMI performs better for the *subdivision* class while PD performs better for the *Fjord* class). Therefore, the effectiveness of the *Recommendation Algorithm* can be further enhanced if the generated rankings are the outcome of the combination of two or more measures.

A limitation of the geospatial approach is that it can be applied only on datasets that contain georeferenced point instances. However, the number of WoD datasets is growing and many datasets contain geographically grounded but not georeferenced instances that can be geo-annotated [29]. Moreover, our approach can support other geometry types (i.e., lines and polygons) with minor changes. Thus, a geospatial approach can be still applied for a significant amount of WoD datasets and classes. In this work, we identified 54 datasets that contain spatial instances and 20,640 spatial classes by parsing only SPARQL Endpoints excluding datasets provided through other means (e.g., RDF files).

7 Future Work and Conclusion

As previously noted, further work needs to be done for the performance optimization of the QuadTree construction algorithm that will include the study of the effect of different parameter settings (e.g., number of classes allowed for triggering a cell split) and experiments on the associated maintenance and complexity costs. Another next step would be to integrate all (or the most effective) spatial measures in a single model (for example, a linear model or a supervised classification model) that would produce enhanced recommendation lists. Finally, a possible future work will examine in depth the strengths and weaknesses of each dataset recommendation approach (i.e., keyword, graph, linkage and spatial) aiming at the implementation of a dataset recommender that will use complementary the above-mentioned approaches in order to improve its overall effectiveness.

To conclude, classes with similar spatial distribution is likely to contain semantically related instances and therefore are relevant for recommending them for Link Discovery. The presented methods, which summarize classes spatial extent (ConvexHull) and spatial distribution (based on a QuadTree) and measure a degree of geospatial relatedness between classes, support this hypothesis and provide high quality ranked lists of recommended classes (up to 62% mean average precision). Moreover, the search space for relevant classes can be reduced up to 99% by applying simple spatial filters. Therefore, the exploitation of the geospatial information in Web of Data datasets can be regarded as a valuable contributor in the dataset recommendation for the Link Discovery domain.

Acknowledgements This research is being supported by the funding program "YPATIA" of University of Aegean.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Adelfio MD, Nutanong S, Samet H (2011) Similarity search on a large collection of point sets. In: Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems. ACM, New York, GIS '11, pp 132–141. https://doi.org/10.1145/2093973.2093992
2. Ballatore A, Bertolotto M, Wilson DC (2014) An evaluative baseline for geo-semantic relatedness and similarity. GeoInformatica 18(4):747–767
3. Ben Ellefi M, Bellahsene Z, Dietze S, Todorov K (2016a) Beyond established knowledge graphs-recommending web datasets for data linking. In: Bozzon A, Cudre-Maroux P, Pautasso C (eds) Web engineering. Springer, Cham, pp 262–279
4. Ben Ellefi M, Bellahsene Z, Dietze S, Todorov K (2016b) Dataset recommendation for data linking: An intensional approach. In: Proceedings of the 13th international conference on the semantic web. latest advances and new domains, vol 9678. Springer, Berlin, pp 36–51
5. Berners-Lee T (2006) Linked data. https://www.w3.org/DesignIssues/LinkedData.html. Last accessed 16 August 2019
6. Bouma G (2009) Normalized (pointwise) mutual information in collocation extraction. In: Proceedings of the Biennial GSCL conference 2009
7. Caraballo AAM, Arruda NM, Nunes BP, Lopes GR, Casanova MA (2014) Trtml—a tripleset recommendation tool based on super-

vised learning algorithms. In: Presutti V, Blomqvist E, Troncy R, Sack H, Papadakis I, Tordai A (eds) The semantic web: ESWC 2014 satellite events. Springer, Cham, pp 413–417

8. Caraballo AAM, Nunes BP, Casanova MA (2016) Drx: A lod dataset interlinking recommendation tool
9. Chapman A, Simperl EPB, Koesten L, Konstantinidis G, Ibáñez-Gonzalez LD, Kacprzak E, Groth PT (2019) Dataset search: a survey. arXiv:abs/1901.00735
10. Das Sarma A, Fang L, Gupta N, Halevy A, Lee H, Wu F, Xin R, Yu C (2012) Finding related tables. In: Proceedings of the 2012 ACM SIGMOD international conference on management of data. ACM, New York, SIGMOD '12, pp 817–828. https://doi.org/10.1145/2213836.2213962
11. Efstathiades C, Belesiotis A, Skoutas D, Pfoser D (2016) Similarity search on spatio-textual point sets. In: EDBT
12. Emaldi M, Corcho Ó, de Ipiña DL (2014) Detection of related semantic datasets based on frequent subgraph mining. In: IESD@ISWC
13. Feliachi A, Abadie N, Hamdi F (2017) An adaptive approach for interlinking georeferenced data. In: Proceedings of the knowledge capture conference. ACM, New York, K-CAP 2017, pp 12:1–12:8
14. Harth A, Hose K, Karnstedt M, Polleres A, Sattler KU, Umbrich J (2010) Data summaries for on-demand queries over linked data. In: Proceedings of the 19th international conference on world wide web. ACM, New York, WWW '10, pp 411–420. https://doi.org/10.1145/1772690.1772733
15. Heath T, Bizer C (2011) Linked data: evolving the web into a global data space. Synth Lect Seman Web Theory Technol 1(1):1–136. https://doi.org/10.2200/S00334ED1V01Y201102WBE001
16. Hecht B, Raubal M (2008) GeoSR: Geographically explore semantic relations in world knowledge. Springer, Berlin, pp 95–113
17. Kanza Y, Kravi E, Safra E, Sagiv Y (2017) Location-based distance measures for geosocial similarity. ACM Trans Web 11(3):17:1–17:32. https://doi.org/10.1145/3054951
18. Kufer S, Henrich A (2014) Hybrid quantized resource descriptions for geospatial source selection. In: Proceedings of the 4th international workshop on location and the web. ACM, New York, LocWeb '14, pp 17–24. https://doi.org/10.1145/2663713.2664428
19. Lehmberg O, Ritze D, Ristoski P, Meusel R, Paulheim H, Bizer C (2015) The mannheim search join engine. Web Semant 35(P3):159–166. https://doi.org/10.1016/j.websem.2015.05.001
20. Leme LAPP, Lopes GR, Nunes BP, Casanova MA, Dietze S (2013) Identifying candidate datasets for data interlinking. In: Daniel F, Dolog P, Li Q (eds) Web engineering. Springer, Berlin, pp 354–366
21. Liu H, Wang T, Tang J, Ning H, Wei D, Xie S, Liu P (2016) Identifying linked data datasets for sameas interlinking using recommendation techniques. In: Cui B, Zhang N, Xu J, Lian X, Liu D (eds) Web-age information management. Springer, Cham, pp 298–309
22. Liu H, Wang T, Tang J, Ning H, Wei D (2017) Link prediction of datasets sameAs interlinking network on web of data. In: 3rd international conference on information management (ICIM), pp 346–352. https://doi.org/10.1109/INFOMAN.2017.7950406
23. Lopes GR, Leme LAPP, Nunes BP, Casanova MA, Dietze S (2013) Recommending tripleset interlinking through a social network approach. In: Lin X, Manolopoulos Y, Srivastava D, Huang G (eds) Web information systems engineering—WISE 2013. Springer, Berlin, pp 149–161
24. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, New York
25. Martins YC, da Mota FF, Cavalcanti MC (2016) Dscrank: a method for selection and ranking of datasets. In: Garoufallou E, Subirats Coll I, Stellato A, Greenberg J (eds) Metadata and semantics research. Springer, Cham, pp 333–344
26. Mehdi M, Iqbal A, Hogan A, Hasnain A, Khan Y, Decker S, Sahay R (2014) Discovering domain-specific public sparql endpoints: a life-sciences use-case. In: Proceedings of the 18th international database engineering and applications symposium. ACM, New York, IDEAS '14, pp 39–45. https://doi.org/10.1145/2628194.2628220
27. Mountantonakis M, Tzitzikas Y (2018) Scalable methods for measuring the connectivity and quality of large numbers of linked datasets. J Data Inf Qual 9(3):15:1–15:49
28. Nentwig M, Hartung M, Ngonga Ngomo AC, Rahm E (2015) A survey of current link discovery frameworks. Semantic Web (Preprint):1–18. http://www.semantic-web-journal.net/system/files/swj1117.pdf
29. Neumaier S, Polleres A (2019) Enabling spatio-temporal search in open data. J Web Semant 55:21–36. https://doi.org/10.1016/j.websem.2018.12.007
30. Ngomo ACN, Auer S (2011) Limes - a time-efficient approach for large-scale link discovery on the web of data. In: IJCAI
31. Nikolov A, d'Aquin M (2011) Identifying relevant sources for data linking using a semantic web index. In: WWW2011 workshop: linked data on the web (LDOW 2011) at 20th international world wide web conference (WWW 2011)
32. Nikolov A, d'Aquin M, Motta E (2012) What should I link to? Identifying relevant sources and classes for data linking. In: Pan JZ, Chen H, Kim HG, Li J, Horrocks I, Mizoguchi R, Wu Z, Wu Z (eds) The semantic web. Springer, Berlin, pp 284–299
33. Röder M, Ngonga Ngomo AC, Ermilov I, Both A (2016) Detecting similar linked datasets using topic modelling. In: Proceedings of the 13th international conference on the semantic web. Latest advances and new domains, vol 9678. Springer, Berlin, pp 3–19
34. Saleem M, Khan Y, Hasnain A, Ermilov I, Ngonga Ngomo AC (2014) A fine-grained evaluation of sparql endpoint federation systems. Semant Web J. https://doi.org/10.3233/SW-150186
35. Schmachtenberg M, Bizer C, Paulheim H (2014a) Adoption of the linked data best practices in different topical domains. In: Mika P, Tudorache T, Bernstein A, Welty C, Knoblock C, Vrandečić D, Groth P, Noy N, Janowicz K, Goble C (eds) The semantic web—ISWC 2014. Springer, Cham, pp 245–260
36. Schmachtenberg M, Bizer C, Paulheim H (2014b) State of the lod cloud 2014. http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/. Last accessed 16 August 2019
37. Schwering A, Raubal M (2005) Spatial relations for semantic similarity measurement. In: Akoka J, Liddle SW, Song IY, Bertolotto M, Comyn-Wattiau I, van den Heuvel WJ, Kolp M, Trujillo J, Kop C, Mayr HC (eds) Perspectives in conceptual modeling. Springer, Berlin, pp 259–269
38. Sherif MA, Ngomo ACN (2017) A systematic survey of point set distance measures for link discovery. Semant Web 9:589–604
39. Sun W, Chou CP, Stacy AW, Ma H, Unger J, Gallaher P (2007) Sas and spss macros to calculate standardized Cronbach's alpha using the upper bound of the phi coefficient for dichotomous items. Behav Res Methods 39(1):71–81. https://doi.org/10.3758/BF03192845
40. Tobler WR (1970) A computer movie simulating urban growth in the detroit region. Econ Geogr 46(sup1):234–240. https://doi.org/10.2307/143141
41. Tummarello G, Cyganiak R, Catasta M, Danielczyk S, Delbru R, Decker S (2010) Sig.ma: live views on the web of data. J Web Semant 8(4):355–364. https://doi.org/10.1016/j.websem.2010.08.003
42. Vidal ME, Castillo S, Acosta M, Montoya G, Palma G (2016) On the selection of sparql endpoints to efficiently execute federated sparql queries. In: Hameurlain A, Kung J, Wagner R (eds) Transactions on large-scale data- and knowledge-centered systems XXV. Springer, Berlin, pp 109–149
43. Vilches-Blázquez LM, Saquicela V, Corcho O (2012) Interlinking geospatial information in the web of data. Springer, Berlin, pp 119–139

44. Volz J, Bizer C, Gaedke M, Kobilarov G (2009) Discovering and maintaining links on the web of data. In: Bernstein A, Karger DR, Heath T, Feigenbaum L, Maynard D, Motta E, Thirunarayan K (eds) The semantic web—ISWC 2009. Springer, Berlin, pp 650–665
45. Wu Z, Palmer M (1994) Verbs semantics and lexical selection. In: Proceedings of the 32nd annual meeting on association for computational linguistics, Stroudsburg, PA, ACL '94, pp 133–138. https://doi.org/10.3115/981732.981751
46. Zhu R, Hu Y, Janowicz K, McKenzie G (2016) Spatial signatures for geographic feature types: examining gazetteer ontologies using spatial statistics. Trans GIS 20(3):333–355. https://doi.org/10.1111/tgis.12232

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.