Computing and Querying Strict, Approximate, and Metrically-Refined Topological Relations in Linked Geographic Data

Blake Regalia[1], Krzysztof Janowicz[1], and Grant McKenzie[2]

[1] STKO Lab, University of California, Santa Barbara, USA
blake@geog.ucsb.edu and janowicz@ucsb.edu
[2] Platial Analysis Lab, McGill University, Montréal, CA
grant.mckenzie@mcgill.ca

Abstract. Geographic entities and the information associated with them play a major role in Web-scale knowledge graphs such as Linked Data. Interestingly, almost all major datasets represent places and even entire regions as point coordinates. There are two key reasons for this. First, complex geometries are difficult to store and query using the current Linked Data technology stack to a degree where many queries take minutes to return or will simply time out. Secondly, the absence of complex geometries confirms a common suspicion among GIScientists, namely that for many everyday queries place-based relational knowledge is more relevant than raw geometries alone. To give an illustrative example, the statement that the White House is in Washington DC is more important for gaining an understating of the city than the exact geometries of both entities. This does not imply that complex geometries are unimportant but that (topological) relations should also be extracted from them. As Egenhofer and Mark put it in their landmark paper on naive geography, *topology matters, metric refines.* In this work we demonstrate how to compute and utilize strict, approximate, and metrically-refined topological relations between several geographic feature types in DBpedia and compare our results to approaches that compute result sets for topological queries on-the-fly.

Keywords: Linked Data, Topology, Geospatial Semantics, GeoSPARQL, Ontology

1 Motivation and Research Contribution

Places and the information associated with them are among the most interlinked types of entities on the global Linked Data cloud (Heath and Bizer, 2011). Within such Web-scale, cross-domain knowledge graphs, places act as pivotal vertices connecting events, people, and objects. Repositories that contain large collections of geographic identifiers are among the most central and densely interlinked hubs on the Linked Data cloud. For instance, *named* places are the second most frequent entities within DBpedia and collectively contribute millions of properties to the dataset, including some of the most common property types such as birthplaces of historic figures and administrative subdivision names.

Nonetheless, the vast majority of geographic identifiers are represented in the simplest of all possible spatial representations, namely point coordinates. While such representation is appropriate for many everyday information retrieval tasks, e.g., finding nearby restaurants, it is not suitable for the plethora of operations performed by scientists, government agencies, and industry professionals using geographic information systems and spatial analysis more broadly. To some extent, these demands could be met by simply providing the more complex geometries, e.g., polylines and polygons, for places in raw form as Linked Data. However, such approaches overlook the key underlying issues. (1) Querying high-resolution geometries, e.g., the areal extent of a river, by using on-demand spatial extensions to triplestores, such as GeoSPARQL, does not scale well over large datasets. (2) Real world applications for complex geometries and semantically empowered queries requires preprocessing steps, e.g., to handle so-called sliver polygons, that are not currently supported by any Linked Data based framework. (3) The ultimate purpose of spatial analysis is often concerned with topological information, e.g., whether a river runs through a city, thereby turning geometries into a "means to an end" for acquiring the topological relation between entities. (4) Finally, the proper geometric representation of real-world entities varies by place type, scale, and task, often leading to unintended consequences when operating on raw, precomputed geometries alone. For instance, representing a state park as a point-feature may be sufficient to get a general sense of its location, but the representation does not support queries for adjacent water bodies. An unintended consequence may be that the centroid of the park is in one county but the extent of the park actually spans two or more counties leading to improper topological results.

With the advent of GeoSPARQL (Perry and Herring, 2012) and other means to perform spatio(temporal) queries (Koubarakis and Kyzirakos, 2010) over Linked Data, complex geometries are becoming more popular across several datasets. The LinkedGeoData project (Stadler et al., 2012), for example, provides different geometry types, such as polygons, extracted from OpenStreetMap. These geometries can be utilized for two types of queries, those that involve or infer topological relations and those that are non-topological such as distances, buffers, patterns, and convex hulls.

Based on the presented argumentation, we conjecture that replacing the simple geometries that dominate knowledge graphs and search engines today with more complex geometries will be of limited use for many everyday applications. Instead, we believe that knowledge graphs and Linked Data more concretely will benefit further from topological relations. One could now argue that such topological relations can be computed using geometries but not the other way around. While this is true in an abstract mathematical sense, it does not hold for actual data. In fact, topological relations between places cannot be easily computed based on geometry alone. While there are many reasons for this (Franklin, 1984;

Ubeda and Egenhofer, 1997), our argument will focus on the role of domain knowledge, vagueness, and uncertainty (Bennett, 2001) and not on computational issues. The fact that simple point geometries are sufficient for the most frequent Point Of Interest (POI) queries has been sufficiently demonstrated by major search and map engines, POI repositories, and place-based social networks, Wikipedia, and so on. Therefore, we will only consider places that are of sufficient spatial extent to result in substantial inaccuracies when modeled as point features alone. Examples include rivers, roads, counties, parks, and so on.

To understand how topology is handled in GIS, it is important to note that data collection, modeling, and preprocessing take about 80% of the entire time budget of a typical GIS project. When data are loaded into a GIS, the analyst uses a sequence of toolboxes to first correct common errors such as so-called *sliver polygons* and then applies domain-specific topological consistency rules.[1] Neither the preprocessing steps nor the domain-specific topology rules are available when computing topological relations on-demand using GeoSPARQL over Linked Data. Also, the datasets used for any given GIS task that involve topological relations are orders of magnitude smaller than querying such relations over Linked Data hubs such as DBpedia. Hence, queries such as finding cities along the Mississippi River or counties that run along state borders cannot be effectively answered over Linked Data today.

Consider the following illustrative example. Lynchburg, Tennessee is a consolidated city-county whose boundaries coincide with Moore County. Using Region Connection Calculus 8 (RCC8) (Cohn et al., 1997), the true topological relation between the city and county should be *equal* however computing the relation using GeoSPARQL returns *partially overlaps*; see Fig. 1. The reason is due in large part to digitization errors, i.e., the double-digitized boundaries problem While differences in granularity are common sources of errors, difficulties arising from uncertainty and vagueness are even more troublesome. Whereas uncertainty stems from lack of precise knowledge, vagueness is caused by intrinsically under-determined concepts that do not have clear borders (Bennett, 2001). For example, the true shape of a city can be determined in theory although measurement accuracy, timeliness (the city may grow or shrink), administrative definitions, and so forth, impact the results. In contrast, the shape of a mountain or forest cannot be exactly determined in practice nor theory as the transition zones between a mountain and a valley, as well as a forest and isolated trees, are *conceptually* vague.

Problem statement: Following Egenhofer and Mark's slogan that *topology matters, metric refines* (Egenhofer and Mark, 1995b), knowledge graphs will benefit from explicit topological relations in addition to (complex) geometries and other place-specific properties. Computing such relations, e.g., using GeoSPARQL, based on geometry alone is not currently possible in the context of Linked Data.

[1]See, for example, the following overview of geodatabase topology rules provided by ArcGIS `http://resources.arcgis.com/en/help/main/10.2/01mm/pdf/topology_rules_poster.pdf`.

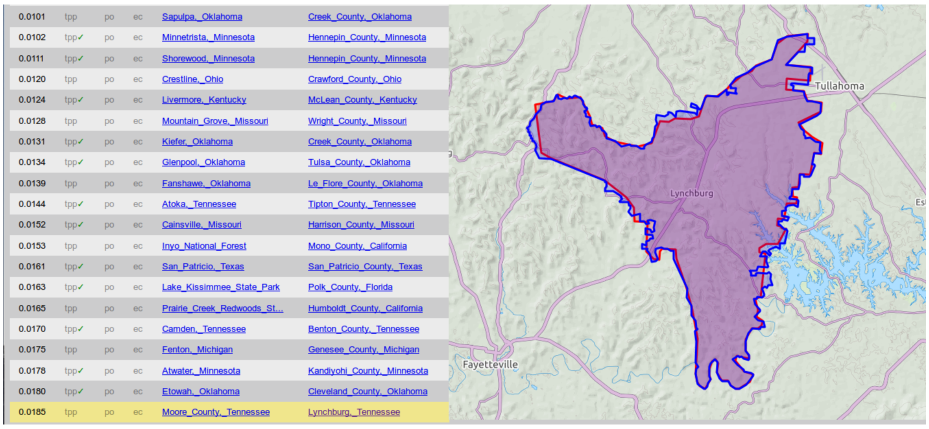

Fig. 1 Lynchburg, Tennessee is a consolidated city-county whose boundaries coincide with Moore County. The expected topological RCC8 relation should be *equal* (**EQ**), however computing the relation solely given the geometries will return *partial overlap* (**PO**).

To address this problem, we propose to combine techniques from GIS and the Semantic Web then demonstrate how to derive strict, approximate, and metrically-refined topological relations, how to use background knowledge in the form of RDF triples that, while strictly speaking are not topological, can be used to infer topological relations, how to define an ontology to distinguish between geographic feature types that have broad boundaries versus those that do not, and finally, how to integrate the aforementioned methods into a multi-layered topological relations framework to enrich DBpedia.

In terms of a bigger picture, this work is about exploring one of three major trade-offs to bring the full Digital Line Graph data from the USGS National Map to the Linked Data cloud. The first trade-off is the decision about which relations to compute on-the-fly and which to materialize (Regalia et al., 2016). For instance, dependent properties such as population densities should be computed if the population count and area are already stored as triples. Similarly, while DBpedia stores select cardinal direction triples, storing all of them would lead to a combinatorial explosion. The work presented here takes a complementary perspective by looking at relations that cannot be easily computed on-the-fly, and, thus, should be precomputed and materialized instead. We will show that queries which include topological relations often cannot be effectively answered using GeoSPARQL. The third tradeoff is about balancing client-side versus server-side queries (Regalia, 2017).

The research contributions of this work are as follows:

- To demonstrate the feasibility of the proposed methods, we present a linked dataset of topological relations derived from the geometries of cities, counties, parks, streams, and roadways for the contiguous United States. We selected these feature types since they cover both strict boundaries (e.g., administrative boundaries) and broad boundaries (e.g., streams) as well as the pairwise relations between regions-to-regions, regions-to-polylines and polylines-to-polylines.

- In addition to the strict topological relations based on a subset of RCC8, we also include approximate topological relations (Clementini and Di Felice, 1997), and additional topological relations with metric refinements (Egenhofer and Dube, 2009). To the best of our knowledge, these extended topological relations have never before been used in the context of Semantic Web research and are neither part of any linked dataset nor ontology.
- We demonstrate how to derive these relations by applying methods known from geographic information systems to features matched between DBpedia and OpenStreetMap. We show how Semantic Web technology can be leveraged to discover latent properties within a heterogeneous geographic dataset by applying topological reasoning. One example would be the creation of a `coastal city` class, defined as a city that has the `(broadly) touches` relation to a feature of the class `ocean`. We will discuss a more complex example about the topological relation of parks and county borders.
- We present example queries based on the resulting dataset and compare them to using GeoSPARQL for qualitative spatial reasoning.
- Finally, we show how our work can help in detecting and clearing erroneous place type definitions in DBpedia based on implausible topological relations.

In this work, we discuss the primary challenges to computing and representing topological relations solely from geometries, demonstrate how to use ontologies and multi-layered topological relations to overcome these challenges, and produce a preprocessed and cleaned dataset of topologically linked places derived from DBpedia and OpenStreetMap.

The remainder of the paper is structured as follows. In Section 2, we describe the process of preparing data collected from DBpedia and OpenStreetMap in order to compute topological relations. In Section 3, we describe how we compute the relations, including for crisp boundaries, broad/approximate boundaries, and metrically-refined topological relations. In Section 4, we provide an overview of the resulting dataset, show a comparison to computing topological relations using GeoSPARQL, and demonstrate the utility of our dataset by example. Finally, we conclude the paper and point to directions for future work.

2 Data Preparation

In this section, we discuss the procedure for constructing a spatially-enabled database in preparation for computing topological relations. The database combines and resolves RDF resources from DBpedia with spatial elements[2] from OpenStreetMap.

[2]From OSM's conceptual data model terminology: `https://wiki.openstreetmap.org/wiki/Elements`

2.1 Data Integration

The goal of this work is to enrich DBpedia with topological relations by producing a dataset of RDF triples. Since DBpedia places are only represented as point coordinates, our first task is to match as many places from DBpedia with their corresponding polygon or polyline geometries in OpenStreetMap. This type of coreference resolution task presents a number of challenges, most notably those discussed by Sehgal et al. (2006) and Ngomo (2012). Existing approaches include the use of string similarity measures (Michalowski et al., 2004) and spatial signatures (Zhu et al., 2016), to name a few. In this paper however, we focus on topological methods and the accuracy of resulting topological relations.

Therefore, we rely on existing meta-level links between the two datasets, i.e., matching normalized Wikipedia and Wikidata URIs through (a) "owl:sameAs" and "foaf:isPrimaryTopicOf" objects from DBpedia triples and (b) "wikidata" and "wikipedia"/"wikipedia:en" tag values from OSM elements in order of precedence. We show an example for Yosemite National Park in Listings 1.1 and 1.2. Approximately 90k OSM elements in North America have such links to DBpedia.

It is important to note that compared to OpenStreetMap which strives for comprehensive geographic coverage, DBpedia exhibits a sparser coverage yet contains a greater depth of information per feature. This is a natural consequence of the fact that Wikipedia, DBpedia's data source, is primarily driven by community members writing articles about topics of societal significance such as cities, national parks, important historic landmarks, and so on. Consequently, coverage is not a primary concern since our goal is to produce an RDF dataset for the Linked Open Data cloud of which DBpedia is the central hub. It's also worth mentioning that the relatively sparser coverage does not jeopardize our ability to compute topology since we are only interested in materializing relations between existing resources when they available. In other words, we envision our approach as being able to adapt to varying degrees of data availability.

```
# http://dbpedia.org/resource/Yosemite_National_Park
dbr:Yosemite_National_Park owl:sameAs wikidata:Q180402 ;
    foaf:isPrimaryTopicOf wikipedia-en:Yosemite_National_Park .
```

Listing 1.1 An example of the meta-level links that exist for Yosemite National Park in an RDF document from DBpedia http://dbpedia.org/data/Yosemite_National_Park.ttl

```
<!-- https://www.openstreetmap.org/relation/1643367 -->
<osm>
    <relation id="1643367">
        <tag k="wikidata" v="Q180402"/>
        <tag k="wikipedia" v="en:Yosemite National Park"/>
        <!-- polygon geometry and other tag nodes... -->
    </relation>
</osm>
```

Listing 1.2 An example of the meta-level links that exist for Yosemite National Park a relation element from OpenStreetMap. In this case, the feature has both links so the Wikidata entity id "Q180402" is used for resolution and the Wikipedia URI is used to validate the resolution.

In order to store the geometries and compute topological relations between all pairs of geographic features, we use the *PostGIS* spatial extension to *PostgreSQL*. We use Overpass[3] to query for all *ways* and *relations* that have a `wikidata`, `wikipedia` or `wikipedia:en` tag. These features are loaded into a spatially-indexed PostgreSQL table with their id, Wikipedia URL suffix, and geometry.

2.2 Entity Selection

In order to derive meaningful strict, approximate, and metrically-refined topological relations requires tuning place-type-specific parameters. For example, the exact buffer radius to use in order to derive a polygon's broad boundary should differ when comparing two cities versus comparing two national parks for the *approximately adjacent* relation, even assuming all polygons are of similar size. In other words, broad boundaries cannot depend on geometry alone. Therefore, we focus our efforts on a subset of features by selecting those of specific place types. We select cities, counties, and parks, which are represented by multipolygon geometries, as well as roadways and streams, which are represented by polyline geometries.

A final collection of places within the contiguous United States have the following essential properties: A DBpedia resource URI, an OpenStreetMap element URI, some geometry ((multi)polygon or polyline), and a place type tag such as city, county, park, roadway, or stream. A composite overview of these are shown in Figure 2.

2.3 Cleaning Digitization Errors

As a first step in deriving topological relations from the noisy geometries we collect from OpenStreetMap, we define a set of metrics that measure various characteristics of the interaction between two geometries. These metrics are initially defined from a top-down perspective and supported through manual inspection of the data. A custom map-enabled interface is used for inspection, providing a broad overview of possible threshold values for identifying and labeling proper topological relations.

Digitization errors are handled through manual exploration of the data in order to identify a conservative threshold that will coerce relations arising from poor geometric alignment into their *correct* relation. For example, intuitively one might expect that the City of Santa Barbara would be completely contained by Santa Barbara County. In a strictly topological sense, however, the two regions in OpenStreetMap partially overlap, as shown in Figure 3. The area of their difference though is only $11.3m^2$, clearly the result of digitization error. In order to help identify such cases, we construct a range of metrics for each relation during the initial computation of strict topological relations. Relations that result in high values for these measure are intended to signify an increased likelihood of

[3]Overpass Query API: https://wiki.openstreetmap.org/wiki/Overpass_API

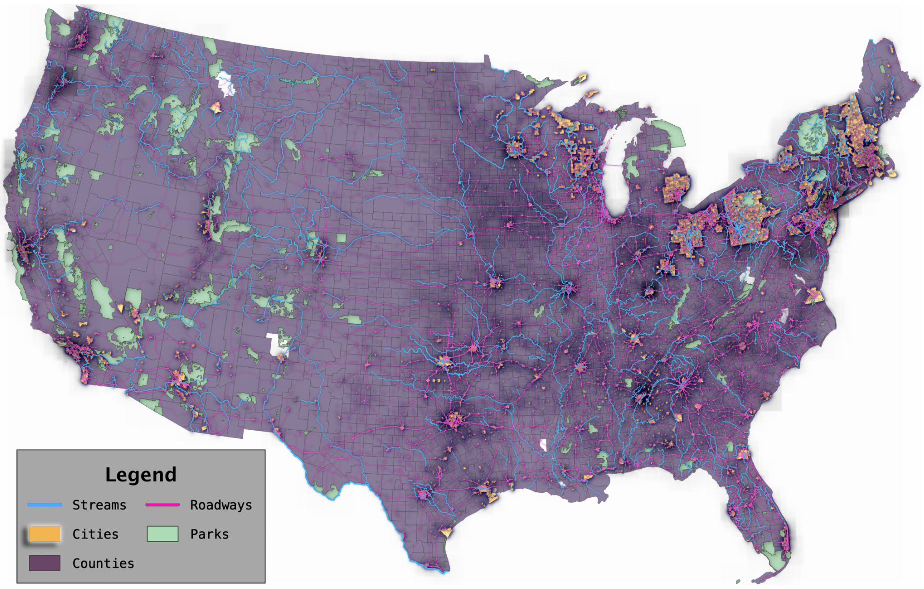

Fig. 2 A map showing a composite overview of all places, categorized by place type, that were matched between DBpedia and OpenStreetMap which we use to compute topological relations. Notice that the 'Cities' symbol uses a drop-shadow effect to reveal the density of small polygon features at this macro-scale.

a digitization error. We designed a visual interface to inspect these measures case-by-case, one at a time, alongside a map view of the geometries (Figure 1).

Egenhofer and Dube (2009) define a set of nine splitting measures for *polygon-overlaps-polygon* relations in order to support metrically refined topology. Here, we apply their Inner Area Splitting (IAS) measure by computing the area of intersection divided by the area of the smaller polygon, i.e., $\frac{Area(L \cap S)}{Area(S)}$. This metric enables us to identify an appropriate threshold to separate cases that should actually be labeled as *tangential proper part* from those that are indeed *overlapping*. This same metric is also used to correct relations erroneously identified as *partial overlaps* to the more suitable *externally connected* (EC). In our analysis, we also measured Inner Traversal Splitting (ITS) and Outer Traversal Splitting (OTS) yet found IAS to be the strongest measure for separating region overlap cases on this dataset.

For all strictly *disjoint* cases, we focus only on those relations that are clearly digitization errors. Adhering to the conceptual neighborhood graph, we can only obtain EC relations from those that start strictly as *disjoint* (DC). Using our custom map-enabled dataset interface, we manually inspected all DC region-region pairs sorted by separation distance in ascending order until reaching cases that were no longer unquestionably disjoint. We then settled on using the conserva-

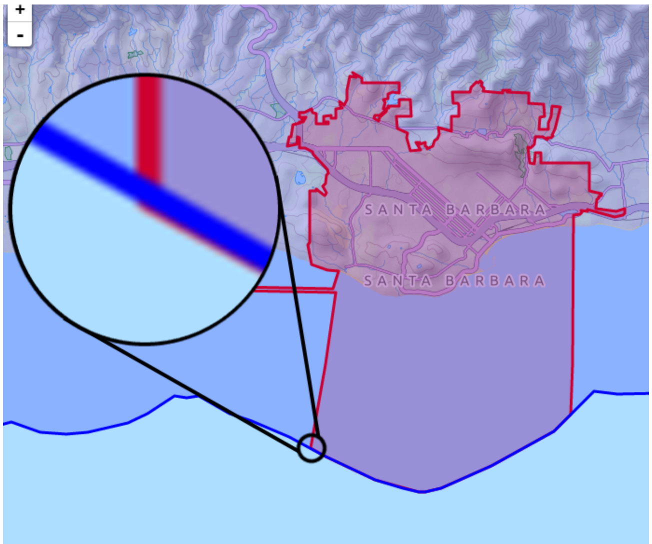

Fig. 3 A render of the geometries for the city of Santa Barbara (red) strictly overlapping Santa Barbara County (blue). The inset zoom bubble shows the $11m^2$ difference of the two geometries.

tive distance threshold value of 20 meters or less between polygon geometries to clean digitization errors by coercing them to the EC relation. In other words, we manually labelled all 375 cases of DC region-region pairs that were corrected to EC. Furthermore, pairs of geometries that are of a distance greater than or equal to 20 meters apart are later used as candidates for the *nearly meets* broad boundary relations. We plan to add more measures such as Expansion Closeness from Egenhofer and Dube (2009), in future work.

3 Computing Topological Relations

In this section we provide details about the selected strict, approximate, and metrically-refined topological relations and their computation.

We start by computing an index of all spatially disjoint features to avoid redundant calculations since every pairwise combination between features must be considered for each topological relation. For instance, to discover that a city and a nearby river are broadly touching, we first need to compute that they are strictly disjoint, yet close enough that their boundaries might overlap. We then proceed by checking topological relations on the remaining pairwise combinations. In fact, we continue this pattern of propagating result sets from each computed topological relation onto the next task in the series to substantially reduce the overall processing time. We provide an example of this technique for the polygon-to-polygon relations procedure in Listing 1.3.

```
# 'cps' stands for 'compute_pairwise_self'
non_interacting := cps_non_interacting(all)
disjoint := non_interacting + cps_disjoint(all - non_interacting)
interacting := all - disjoint
touches := cps_touches(interacting)
intersecting := interacting - touches
overlaps := cps_overlaps(intersecting)
within := cps_within(intersecting - overlaps)
tangential_proper_part := cps_tpp(within)
non_tangential_proper_part := within - tangential_proper_part
```

Listing 1.3 Psuedocode summarizing the procedure for computing the polygon-to-polygon topological relations in series by reusing and subtracting results sets from previous computations, an application of the conceptual neighborhood graph. Notice in this Listing, we use "+" to represent the union of two sets and "-" to represent the relative complement.

3.1 Strict Topological Relations

As opposed to relations between features with broad boundaries, i.e., *approximate* relations, we use the term *strict* to refer to relations about polygons with crisp boundaries and polylines.

Egenhofer and Franzosa (1991) initially defined a framework for the description of topological spatial relations based on the intersections of boundaries and interiors between two sets in $\mathbb{R}^2$. Clementini et al. (1993) extended the 9-Intersection Model (Egenhofer et al., 1993) for topological interactions between spatial regions with the Dimensionally Extended 9-Intersection Model (DE-9IM). Cohn et al. (1997) provide a family of first-order logical calculi known as Region Connection Calculus which treats spatial regions as primitives in order to support reasoning about spatial entities with *connections*. Most notably, RCC8 is a set of eight relations that are jointly exhaustive and pairwise disjointed.

In RCC8, TPPi and NTPPi are inverse relations of TPP and NTPP, respectively. Consequently, the inverse relations are reserved to be inferenced by the RDF reasoner during query execution. In fact, we omit materializing any triples that would be handled by basic reasoning on inverse properties and transitive properties. We also do not materialize *disjoint* relations as this is not only infeasible from a storage perspective but also unnecessary for operating under the Open World Assumption (OWA). Because Linked Data operates under the OWA, dataset publishers may choose to exclude any sets of relations without introducing inconsistencies among their enriched dataset. In total, for region-region relations, we compute *equals* (EQ), *externally connected* (EC), *partially overlapping* (PO), *tangential proper part* (TPP) and *non-tangential proper part* (NTPP).

For strict topological relations between line-region, we compute *touches* (TCH), *passes through* (PTH), and *includes* (INC) (Egenhofer and Mark, 1995a; Formica et al., 2012).

3.2 Approximate Topological Relations

Conceptually there is a disconnect between what is clearly a strict relation and an *approximate* relation. Approximate topological relations (Clementini and Di Felice, 1997) are used to describe broad boundaries (Du et al., 2008) between spatial features. For example, a river may border a city on one side, but topologically the river does not coincide with the border in all sections; rather it *approximately* follows part of the city boundary before continuing on. In such cases, one can argue there is a topological relationship between the two features as the concrete geometric representations of features and their accuracy depend on time, scale, measurement conventions, and so forth. This is particularly the case when both fiat and bona fide boundaries are involved (Smith and Mark, 1998). However, what exactly constitutes a *broad boundary* compared to two features simply being *nearby* requires further exploration.

To determine the radius by which to buffer a polygon's boundary, the 0.05 percentile of the cumulative distribution function of ordered minimum distances between pairs is used as the maximum *broad boundary* measure. We then multiply this by the isoperimetric quotient shown in Equation 1 to account for the observation that features which have very specific (fiat or bona fide) boundaries can be thought of as having a lesser degree of uncertainty as compared to those features that have simple shapes/boundaries.

$$IQ = \frac{4\pi A}{p^2} \tag{1}$$

Finally, the geometric boundaries for each feature pair in our set of disjoint relations are buffered using the above approach. Those feature pairs whose buffered boundaries intersect get assigned an approximate topological relation. For approximate topological relations, we compute *nearly contains* (nCt) and *nearly equals* (nE) for region-region relations, and *nearly meets* (nM) for both polyline-region and region-region relations.

3.3 Metrically-Refined Topological Relations

Here we briefly explain the concept of metrically-refined topological relations and then provide an explanation our four relations: *mostly within* (mW), *barely touches* (bT), *connects* (CON), *runs along* (RAL), and *runs alongside* (RAS).

Metrically-refined topology opens the door to a wide range of potential relations that distinguish more detail about relations between spatial entities than purely qualitative topological ones (Egenhofer and Dube, 2009). This includes predicates that may be conceptually vague and difficult to represent especially as a binary relation. Therefore, we attempt to capture the semantics of concepts that are obvious to human perception, such as a highway running along the ocean, even if they are occasionally inconsistent from a strictly topological point of view (Egenhofer and Mark, 1995b) .

To start off with a straightforward demonstration of metrically-refined topology, we refine the EC relation for region-region by defining *barely touches* (bT) as when the length of the boundary connection is less than $10m$. Although this threshold is not data-driven, we emphasize that the primary goal of metrically-refined topological relations is to provide some meaningful distinction to users, oftentimes allowing domain experts to transparently impose a top-down perspective on the refined relations.

Next, we refine the PO relation for region-region by defining *mostly within* (mW) as when the area of the intersection is greater than or equal to 80% of the smaller polygon's area. This metric is based on the Inner Area Splitting (IAS), one of the nine splitting measures for *region-overlaps-region* relations from Egenhofer and Dube (2009).

For relations involving polylines, we measure the area of intersection between the buffered regions of the two features. If X_D is the minimum bounding diameter of polyline X in meters, we define the buffer radii X_R as $\ln(X_D)$. The buffered polygon X_B is then used to calculate a polyline's interactions with other buffered features for the RAL and RAS relations. In Equation 2, we define the inequality for T, the threshold value for which the *runs along* RAL relation holds between two polylines X and Y. The *runs alongisde* relation applies a similar metric to line-region DC relations by using the buffered boundary and buffer radius of each feature.

$$\frac{Area(X_B \cap Y_B)}{(min(X_D, Y_D))^2} \geq T \tag{2}$$

For the CON relation, we select cases that approximately match the 16-intersection matrix code strings `0*0**0*10*11*111` (e) and `0*1**0*20*02*111` (h) for polyline-polyline relations from Formica et al. (2018), with a relaxation of $\mathcal{E} = 10m$ as the maximum endpoint 'snapping' distance for which a polyline's endpoint is allowed to move in order to intersect the other polyline. This threshold value was selected following the same process described in Section 2.3.

4 Application and Evaluation

In total, we produce 120,681 distinct RDF statements covering various topological relations among features of the selected place types within the contiguous United States. We provide a breakdown of these relations for polygons-to-polygons in Table 2, polylines-to-polygons in Table 3, and polylines-to-polylines Table 4. Next, we demonstrate how to use this dataset to correct place-type classification errors in DBpedia, validate existing adjacency relations in DBpedia, and perform topological queries over Linked Data. We also compare the performance of our materialized relations to querying them on-the-fly using GeoSPARQL.

4.1 DBpedia Error Correction

We compute topological relations between *all* combinations of features, regardless of their place type. However, certain place type combinations should exclude some topological relations by virtue of their ontological axiomatization. For example, two administrative regions of the same class cannot overlap by definition, so no county should ever be contained by another county. Our experiment yields cases that would appear to violate such rules, including the 10 combined county-county proper part relations, the 239 combined city-city proper-part relations, and the 48 city-city partial overlaps relations; see tables 2-4. Manual inspection of these cases reveals that the features involved with these relations are in fact misclassified by DBpedia. Namely, the 10 county-county relations involve places that are actually cities, and the 287 city-city relations mostly involve places that are not cities but actually a variety of place types including cemeteries, airports, buildings, and so on. Many of these DBpedia resources also include `rdf:type` relations to their proper classes but DBpedia does not prevent the aforementioned class violations, e.g., by performing validation on the TBox statement that `Airport` and `City` are disjoint classes. Defining such disjointness axioms, however, for all place types combinations a-priori is not feasible due to many cases that can arise in reality such as cities spanning two counties. The same is true for constraints in the form of SHACL shapes.

It is worth noting that additional complications can arise from the fact that NTTP and EC can be easily confused both in terms of geometric errors and conceptually. For instance, one could naively assume that the village of Birmingham, Missouri is inside (NTTP) of Kansas City, Missouri while, in fact, it is entirely surrounded by it (EC). In contrast, a city is really contained by a county and not externally connected to it. Put differently, the area of Kansas City is determined by its polygon's area minus the holes represented by inner rings, while a city inside a county does not form such an inner ring. In everyday language, however, we typically do not make such distinctions.

4.2 Validating DBpedia's Adjacency and Partonomy Relations

While DBpedia itself does not aim at providing any robust topological relations between places, there do exist avenues for structured and semi-structured data from Wikipedia to make their way into topologically significant relations in DBpedia. For example, the primary cardinal direction relations, `dbp:north`, `dbp:east`, `dbp:south` and, so on, are generated via natural language processing on Wikipedia article abstracts, as well as from a special "Adjacent Communities" wiki template[4]. Ostensibly, these cardinal direction relations in DBpedia encode some meaningful topological relation, namely adjacency, between places. However, as one might suspect, relations are sometimes made to well-known places that are not remotely adjacent simply because they serve as a geographic reference or are in some way significant to the history or function of a place. For instance, Flint, MI has a southwest relation to Chicago, IL even though the two cities are more than $350km$ apart. Nonetheless, triples that make use of such cardinal direction relations offer an opportunity to deploy our materialized topological dataset for comparison.

[4] https://en.wikipedia.org/wiki/Template:Adjacent_communities

In fact, for each of the $82,973$ distinct combinations between places that interact topologically according to our dataset, we query DBpedia for all relations that exist between each pair then rank the set of involved predicates by the number of times they appear in a triple. As we can see in Figure 4, cardinal direction relations make up nearly half of all relations, followed by nearly a third belonging to the `dbo:isPartOf` predicate. This allows us to assess which cardinal directions in DBpedia coincide with topological relations and which do not. As we have demonstrated in previous work (Regalia et al., 2016), approximately 33% of the cardinal direction relations in DBpedia are defective and many other require additional information about the involved uncertainties to become reproducible. We compare all EC, TPP, and NTPP triples and find a majority of statements from DBpedia to be potentially accurate, see Figure 5. Based on the results shown above, such cases should be replaced with topological relations instead, particularity if they have been extracted from Wikipedia's adjacency template.

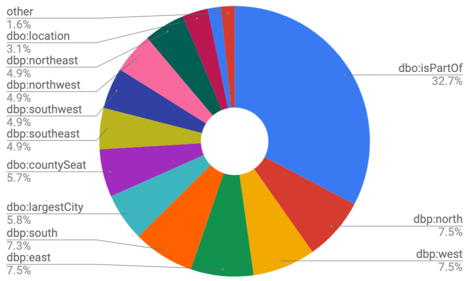

Fig. 4 Relative frequencies of the most common predicates that relate two places to each other on DBpedia, excluding `dbo:wikiPageWikiLink`, for all features that exhibit any topological interaction within our dataset. Collectively, cardinal direction relations constitute nearly 50% of all such triples.

4.3 Relation to GeoSPARQL Queries

A key utility of our resulting dataset is to support topological queries over Linked Data. Given that users can already perform topological queries over Linked Data using GeoSPARQL, in this subsection, we demonstrate the shortcomings of computing topology *on-the-fly*, i.e., in response to queries, illustrate the limitations of using purely crisp boundary topology, and show why the Web of Linked Data needs *cleaned* geometry data and *precomputed* topology encoded with domain knowledge, e.g., for applying the correct relations.

Consider, for example, a query for *how many other counties does each county share a border with?*; shown in Listings 1.4 and 1.5. Using GeoSPARQL, we are able to obtain $3,074$ results in 176 seconds, compared to our approach which yields $3,080$ results in about 9 seconds. Both queries run on the same *cold*, i.e., uncached, triplestore. The difference in performance is expected since the

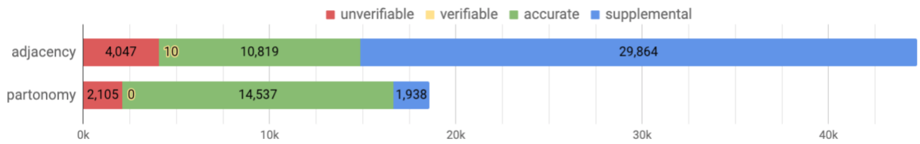

Fig. 5 Stacked bar chart showing how many place pair relations are (a) **unverifiable** based on their absence from our materialized dataset, which suggests potentially inaccurate topological triples on DBpedia, (b) **verifiable** based on their presence in both datasets *with the stipulation* that the topological relation(s) observed in our dataset does not align with the topological relation inferred from DBpedia, (c) **accurate** based on their presence in both datasets *and* the condition that the topological relations align, which supports the topological accuracy of such triples on DBpedia, and (d) **supplemental** based on their absence from DBpedia, which demonstrates the volume of our contribution towards enhancing the LOD cloud. Cardinal direction relations are represented by the *adjacency* label and `dbo:isPartOf` by *partonomy*.

GeoSPARQL approach must compute topology on-the-fly whereas ours is already materialized.[5]

However, this bordering counties example was carefully chosen in order to be able to compare our approach against GeoSPARQL since most other interesting use cases are simply unfeasible for a GeoSPARQL triplestore to handle on-the-fly, i.e., they either timeout or run out of memory, due to the computational cost for each topological relation combined with the large number of pairwise combinations between geometric features in such a dataset. To illustrate, the *city-touches-city* relation we (pre)computed for our dataset took over 12 hours to running on 56 2.1 GHz cores in parallel, while other relations, such as *road-nearlyMeets-road*, took more than 35 hours, combining topological and metric queries in PostGIS.

On the other hand, there are also discrepancies between the two result sets. Out of the $3,074$ counties that both result sets have in common, our approach finds between 1 and 4 **additional** bordering counties in 42 cases where GeoSPARQL does not register EC relations due to sliver polygons. Even more compelling, our approach returns 6 results that do not appear at all in the GeoSPARQL result set due to the fact that their geometries do not exhibit perfectly precise common boundaries with adjacent features. For instance, due to sliver polygons that are imperceptible to the human eye, GeoSPARQL finds 0 bordering counties for Houston County, Georgia, and, thus, it is not included in the result set, whereas our approach yields all 8 bordering counties; see Figure 6. Furthermore, our dataset also materializes a supplementary *barely touches* relation to one of Houston County's bordering counties, Crawford County, Georgia,

[5]Hence, this experiment should not be confused for a runtime performance evaluation but is supposed to demonstrate the feasibility (or lack thereof) of computing with complex geometries on-the-fly.

in order to metrically refine the EC relation. In total, GeoSPARQL failed to capture 78 EC relations between counties.

Finally, querying the broader Web of Linked Data for topological relations by computing them on-the-fly will, at some point, inevitably involve geometries combined from heterogeneous datasets, e.g., by using federated querying or Linked Data aggregators such as the LOD Laundromat[6]. However, this approach to computing topology is fraught with limitations and potential errors due to *dirty* geometries, e.g., the fact that no two sources will digitize the exact same boundaries, and misaligned ontological concepts due to different understandings or modeling decisions about *place types*.

```
# Using GeoSPARQL
select ?countyA ?borderingCounties {
    select ?countyA (count(?countyB) as ?borderingCounties) {
        ?countyA a experiment:County ; geosparql:hasGeometry ?geomA .
        ?countyB a experiment:County ; geosparql:hasGeometry ?geomB .
        filter(geof:sfTouches(?geomA, ?geomB))
    } group by ?countyA
} order by desc(?borderingCounties)
```

Listing 1.4 Query for all bordering counties using GeoSPARQL's *extensible value testing* function `geof:sfTouches`, which computes the **EC** topological relation on-the-fly.

```
# Using our precomputed topological dataset
select ?countyA ?borderingCounties where {
    select ?countyA (count(?countyB) as ?borderingCounties) {
        ?countyA a experiment:County . ?countyB a experiment:County .
        { ?countyA agt:touches ?countyB }
        union { ?countyB agt:touches ?countyA }
    } group by ?countyA
} order by desc(?borderingCounties)
```

Listing 1.5 Query for all bordering counties using our `agt:touches` predicate, which represents the **EC** topological relation that was materialized by precomputing topology for all features.

4.4 Topological Queries over Linked Data

There is another benefit to precomputing and materializing topological relations for use in Web-scale knowledge graphs that is less obvious than performance trade-offs and scalability. The fact that topological relations are materialized as object properties in RDF allows users to define custom axioms, such as class assertions, in order to perform topological and subclass reasoning on a geographic dataset. In this section, we provide an example, created to reflect a potential scenario from our dataset, that illustrates the capabilities of topological reasoning as it applies to Linked Data.

In this example, we wish to create a class that identifies parks which would require a traveler to entirely cross through the interior of one or more counties in order to reach the park's region from a starting location on the containing state's

[6] http://lodlaundromat.org/

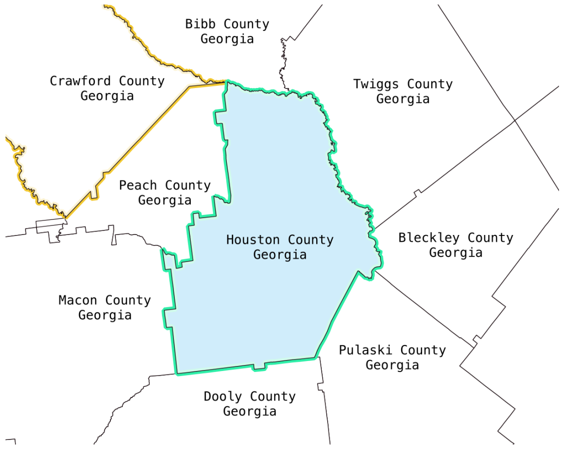

Fig. 6 Houston County, Georgia shares a border with 8 counties, none of which are captured by GeoSPARQL's `geof:sfTouches` topological operator function due to tiny sliver polygons. Our approach also materializes the `agt:barelyTouches` relation to Crawford County, Georgia.

boundary. Conceptually, we assume that the traveler cannot traverse along the zero-width, one-dimensional edges of county boundaries and must therefore be within exactly one county at any given location. We define the axioms in Equation 3, starting with the assertion that counties cannot overlap with, nor be within, other counties. From the jointly exhaustive and pairwise disjoint set of relations from RCC8, this implies that counties must either be DC or EC to other counties. We then apply a similar assertion to US states, followed by the axiom for non-tangential counties (`NTC`), which defines counties that are NTPP to a state. Finally, we define our target class, `ParksInNTC` which identifies parks that either only have PO relations to `NTC`s or are a NTPP of an `NTC`.

$$\begin{aligned} \text{County} \sqcap (\exists\ \text{PO.County} \sqcup \exists\ \text{NTPP.County} \sqcup \exists\ \text{TPP.County}) &\sqsubseteq \bot \\ \text{State} \sqcap (\exists\ \text{PO.State} \sqcup \exists\ \text{NTPP.State} \sqcup \exists\ \text{TPP.State}) &\sqsubseteq \bot \\ \text{NTC} \equiv \text{County} \sqcap \exists\ \text{NTPP.State} & \\ \text{ParksInNTC} \equiv \text{Park} \sqcap (\forall\ \text{PO.NTC} \sqcup \exists\ \text{NTPP.NTC}) & \end{aligned} \tag{3}$$

Another practical use for Semantic Web technology on topological relations can be to support question answering systems, which typically involve conceptually vague relations to begin with, such as *nearby*. In this example scenario, we translate the question *are there important figures who were born in one city along the Mississippi River and died in a different city along the Mississippi River, and if so, who are they and which cities where involved?* into a query. Here, we attempt to model the relation *along* with a metrically-refined topological relation, *runs alongside* (RAS), defined in Section 3.3, in an effort to implement a *naive geographic model* which Egenhofer and Mark (1995b) propose as the first stage in a feedback loop that ideally aligns formal models with intuitive human perception. The SPARQL query is shown in Listing 1.6, while the result is illustrated graphically in Figure 7.

```
select ?person ?placeBorn ?placeDied where  {
    ?placeBorn a :City .  ?placeDied a :City .

    dbr:Mississippi_River ?interactsA ?placeBorn .
    values ?interactsA { agt:touches agt:crosses agt:nearlyMeets }

    dbr:Mississippi_River ?interactsB ?placeDied .
    values ?interactsB { agt:touches agt:crosses agt:nearlyMeets }

    filter(?placeBorn != ?placeDied)

    service <http://dbpedia.org/sparql/> {
        ?person a dbo:Person ;
            dbo:birthPlace ?placeBorn ;
            dbo:deathPlace ?placeDied .
    }
}
```

Listing 1.6 SPARQL query for persons who were born in a city along the Mississippi River and died in a different city along the Mississippi River using a *Federated Query* to combine our topological dataset with DBpedia's knowledge graph.

Fig. 7 A map of the trajectories of persons who were born in a city along the Mississippi River and died in a different city along the river.

5 Conclusions and Further Work

Publishing massive geographic datasets with complex geometries as Linked Data requires balancing several trade-offs. One family of trade-offs is concerned with the question of which properties to compute on-the-fly, i.e., during query time, and which to store in pre-computed form. In this work, we argued why topological relations (and queries involving them) often cannot be computed during query time, despite being supported in theory by GeoSPARQL. Additionally, GeoSPARQL and related approaches only support a subset of relationships relevant for everyday queries. Following Egenhofer and Mark's slogan that *topology matters, metric refines*, we compute polygon-polygon, polygon-polyline, and polyline-polyline topological relations for several feature types such as cities, parks, and roadways in DBpedia. As DBpedia does not contain complex geometries but merely points, we derive the geometries from aligning DBpedia entities with OpenStreetMap. On top of strict topological relations, here RCC8, we also compute approximate and metrically-refined relations. Interestingly, both approximate and metrically-refined relations have not been studied in the geospatial semantics literature before and no ontologies or datasets have been published. We present a variety of interesting findings such as how to detect classification errors in DBpedia and how to validate existing adjacency rations. Finally, we give examples for queries enabled by our approach and compare their runtime and results with GeoSPARQL. From a big picture perspective, our work contributes to finding the right balance between cases where complex geometries should be made available as Linked Data and cases where providing point data enriched by topological relations computed based on these complex geometries is sufficient. We provide the source code to our custom computational framework at https://github.com/blake-regalia/awesemantic-geo, along with a live SPARQL endpoint of the materialized dataset which can be queried using a web interfae at http://yasgui.org/short/Lp1v0cYL4.

Future work will focus on computing topological relations for the full USGS Digital Line Graph dataset and publishing them as Linked Data. We also hope to integrate the current dataset with DBpedia. We also aim at developing a full ontology for strict, approximate, and metrically-refined relations, an addition to the subset presented in the current work. Finally, as it is difficult to find a context-independent definition for the range of broad boundaries and even more so for metrically-refined topological relations, we plan to introduce a second provenance graph, e.g., using PROV-O with additional axioms that model uncertainty, that enables users of topologically linked data to understand the individual design decisions that went into creating the data.

Code	Types	Description
	L	Refers to (Multi)Polyline geometry types.
	G	Refers to (Multi)Polygon geometry types.
	E	Refers to *either* of the two aforementioned geometry types.
		Crisp Boundary Relations for G/G pairs – RCC8 (Cohn et al., 1997)
DC	G/G	Disconnected
EC	G/G	Externally Connected
PO	G/G	Partially Overlaps
EQ	G/G	Equals
TPP/i	G/G	Tangential Proper Part ∪ Tangential Proper Part Inverse
NTPP/i	G/G	Non-Tangential Proper Part ∪ Non-Tangential Proper Part Inverse
		Crisp Boundary Relations for L/G pairs – as used by Formica et al. (2012).
TCH	L/E	Touches
PTH	L/G	Passes Through
INC	E/L	Inclusion
		Crisp Boundary Relations for L/L pairs Formica et al. (2018).
CRS	L/L	Crosses
TCS	L/L	Touch Crosses $\subset TCH$
		Broad Boundary Relations – as defined by Clementini and Di Felice (2001).
nM	E/G	Nearly Meets $\subset DC$
nCt	G/G	Nearly Contains $\subset PO$
nE	G/G	Nearly Equals $\subset (PO \cup TPP/i \cup NTTP/i)$
		Metrically-Refined Topological Relations
mW	G/G	Mostly Within $\subset PO$: the area of intersection is greater than or equal to 80% of P_1's area.
bT	G/G	Barely Touches $\subset EC$: the spheroidal length of the intersecting boundary is less than $10m$.
RAL RAS	L/E	Runs Along (L/L), Runs Alongside (L/G): the area of intersection between the features' broad boundary buffers is greater than some threshold value as described in Section 3.2.
CON	L/L	Connects $\subset TCH$: at least one of the points where the polylines intersect is colocated with one of the points that either polyline starts or ends.

Table 1 Topological operator codes as defined by related works as well as our custom *metrically-refined* operator codes. P_1 refers to the polygon with lesser area and P_2 the polygon with greater area.

										avg. area of...	
region-region	**EQ**	**EC**	**PO**	**TPP/i**	**NTPP/i**	**nE**	**nM**	**nCt**	**mW**	**bT**	*smaller polygon*	*larger polygon*
park-park	1	220	9	10	49	0	84	0	3	4	$477km^2$	$3,952km^2$
park-city	0	283	160	79	740	0	120	14	47	291	$22km^2$	$617km^2$
park-county	0	516	512	135	1,645	0	15	1	74	439	$411km^2$	$4,971km^2$
city-city	0	11,827	48	58	189	0	386	0	20	27	$65km^2$	$170km^2$
city-county	1	6,768	1,046	3,397	12,496	0	84	5	280	880	$40km^2$	$2,694km^2$
county-county	0	9,117	0	1	9	0	25	0	0	0	$2,048km^2$	$3,302km^2$

Table 2 Number of region-to-region relations materialized for each place type combination by row, and each topological relation by column using codes defined in Table 1.

							avg. length/area of...	
polyline-region	**PTH**	**TCH**	**INC**	**nM**	**bT**	**RAS**	*polyline*	*polygon*
road-park	137	984	155	13,928	10	11	$316km$	$1,169km^2$
road-city	3,072	17,302	3,303	19,528	106	100	$425km$	$137km^2$
road-county	7,041	5,597	3,751	4,579	213	9	$383km$	$2,739km^2$
stream-park	156	220	123	1,303	3	5	$293km$	$4,180km^2$
stream-city	708	2,516	285	2,973	106	382	$408km$	$258km^2$
stream-county	1,502	1,491	1,718	828	118	241	$418km$	$4,221km^2$

Table 3 Number of polyline-to-region relations materialized for each place type combination by row, and each topological relation by column using codes defined in Table 1.

						avg. length of...	
polyline-polyline	**CRS**	**TCS**	**CON**	**nM**	**RAL**	*shorter polyline*	*longer polyline*
road-road	9,861	2	658	100,790	65	$20km$	$127km$
road-stream	4,109	0	84	7,922	94	$79km$	$556km$
stream-stream	12	0	237	4,573	2	$5km$	$24km$

Table 4 Number of polyline-to-polyline relations materialized for each place type combination by row, and each topological relation by column using codes defined in Table 1.

Bibliography

Bennett, B., 2001. What is a forest? on the vagueness of certain geographic concepts. Topoi 20 (2), 189–201.

Clementini, E., Di Felice, P., 1997. Approximate topological relations. International journal of approximate reasoning 16 (2), 173–204.

Clementini, E., Di Felice, P., 2001. A spatial model for complex objects with a broad boundary supporting queries on uncertain data. Data & Knowledge Engineering 37 (3), 285–305.

Clementini, E., Di Felice, P., Van Oosterom, P., 1993. A small set of formal topological relationships suitable for end-user interaction. In: International Symposium on Spatial Databases. Springer, pp. 277–295.

Cohn, A. G., Bennett, B., Gooday, J., Gotts, N. M., 1997. Qualitative spatial representation and reasoning with the region connection calculus. GeoInformatica 1 (3), 275–316.

Du, S., Qin, Q., Wang, Q., Ma, H., 2008. Reasoning about topological relations between regions with broad boundaries. International Journal of Approximate Reasoning 47 (2), 219–232.

Egenhofer, M. J., Dube, M. P., 2009. Topological relations from metric refinements. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, pp. 158–167.

Egenhofer, M. J., Franzosa, R. D., 1991. Point-set topological spatial relations. International Journal of Geographical Information System 5 (2), 161–174.

Egenhofer, M. J., Mark, D. M., 1995a. Modelling conceptual neighbourhoods of topological line-region relations. International journal of geographical information systems 9 (5), 555–565.

Egenhofer, M. J., Mark, D. M., 1995b. Naive geography. In: International Conference on Spatial Information Theory. Springer, pp. 1–15.

Egenhofer, M. J., Sharma, J., Mark, D. M., 1993. A critical comparison of the 4-intersection and 9-intersection models for spatial relations: formal analysis. In: Autocarto Conference. ASPRS American Society for Photogrammetry and Remote Sensing ASPRS.

Formica, A., Mazzei, M., Pourabbas, E., Rafanelli, M., 2012. A 16-intersection matrix for the polygon-polyline topological relation for geographic pictorial query languages. In: International Conference on Availability, Reliability, and Security. Springer, pp. 302–316.

Formica, A., Mazzei, M., Pourabbas, E., Rafanelli, M., 2018. Approximate answering of queries involving polyline–polyline topological relationships. Information Visualization 17 (2), 128–145.

Franklin, W. R., 1984. Cartographic errors symptomatic of underlying algebra problems. In: International Symposium on Spatial Data Handling, Zurich, Switzerland. Vol. 286.

Heath, T., Bizer, C., 2011. Linked data: Evolving the web into a global data space. Synthesis lectures on the semantic web: theory and technology 1 (1), 1–136.

Koubarakis, M., Kyzirakos, K., 2010. Modeling and querying metadata in the semantic sensor web: The model strdf and the query language stsparql. In: Extended Semantic Web Conference. Springer, pp. 425–439.

Michalowski, M., Ambite, J. L., Thakkar, S., Tuchinda, R., Knoblock, C. A., Minton, S., 2004. Retrieving and semantically integrating heterogeneous data from the web. IEEE Intelligent Systems 19 (3), 72–79.

Ngomo, A.-C. N., 2012. On link discovery using a hybrid approach. Journal on Data Semantics 1 (4), 203–217.

Perry, M., Herring, J., 2012. OGC GeoSPARQL - a geographic query language for rdf data. OGC Implementation Standard. Sept .

Regalia, B., 2017. Decentralizing the persistence and querying of rdf datasets through browser-based technologies .

Regalia, B., Janowicz, K., Gao, S., 2016. Volt: a provenance-producing, transparent sparql proxy for the on-demand computation of linked data and its application to spatiotemporally dependent data. In: International Semantic Web Conference. Springer, pp. 523–538.

Sehgal, V., Getoor, L., Viechnicki, P. D., 2006. Entity resolution in geospatial data integration. In: Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems. ACM, pp. 83–90.

Smith, B., Mark, D. M., 1998. Ontology and geographic kinds .

Stadler, C., Lehmann, J., Höffner, K., Auer, S., 2012. LinkedGeoData: A core for a web of spatial open data. Semantic Web 3 (4), 333–354.

Ubeda, T., Egenhofer, M. J., 1997. Topological error correcting in gis. In: International Symposium on Spatial Databases. Springer, pp. 281–297.

Zhu, R., Janowicz, K., Yan, B., Hu, Y., 2016. Which kobani? a case study on the role of spatial statistics and semantics for coreference resolution across gazetteers. In: International Conference on GIScience Short Paper Proceedings. Vol. 1.