Template-Based Question Answering over Linked Geospatial Data*

D. Punjani[1], K. Singh[2], A. Both[3], M. Koubarakis[1], I. Angelidis[1], K. Bereta[1], T. Beris[1], D. Bilidas[1], T. Ioannidis[1], N. Karalis[1],C. Lange[2], D. Pantazi[1], C. Papaloukas[1], G. Stamoulis[1]
[1] Dept. of Informatics and Telecommunications, National and Kapodistrian University of Athens
{dpunjani, koubarak}@di.uoa.gr,
[2] Fraunhofer IAIS and University of Bonn
kuldeep.singh@iais.fraunhofer.de, langec@cs.uni-bonn.de,
[3] DATEV eG, Germany
contact@andreasboth.de

ABSTRACT

Large amounts of geospatial data have been made available recently on the linked open data cloud and on the portals of many national cartographic agencies (e.g., OpenStreetMap data, administrative geographies of various countries, or land cover/land use data sets). These datasets use various geospatial vocabularies and can be queried using SPARQL or its OGC-standardized extension GeoSPARQL. In this paper we go beyond these approaches to offer a question answering service on top of linked geospatial data sources. Our system has been implemented as re-usable components of the Qanary question answering architecture to provide benefits for future research tasks. We give a detailed description of the architecture of the system, its underlying algorithms and its evaluation using a set of 201 natural language questions.

CCS CONCEPTS

• **Information systems → Geographic information systems**; **Web search engines**; **Question answering**;

KEYWORDS

linked geospatial data, DBpedia, General Administrative Divisions dataset (GADM), OpenStreetMap

1 INTRODUCTION

The number of data sources in private environments, enterprises, and the Web is increasing continuously. This circumstance also increases the effort of making data accessible. One important means of making data accessible is *question answering (QA)*, which provides a natural language interface for common users to express their information needs [15]. Users commonly pose questions or information requests with a *geospatial dimension* to search engines, e.g., “Christmas market in Germany”, “Schools in London”, “Is there a Macy’s near Ohio?”, “Which countries border Greece?”. Answering such questions or information requests requires data that has a geospatial dimension as well.

Geospatial or *geographic knowledge* has been studied for many years by researchers in Geography, Geographic Information Systems (GIS), Geographic Information Retrieval (GIR), Databases, Artificial Intelligence and the Semantic Web, and there is a wealth of research results concerning representation, querying and inference for geographic knowledge. In GIS terminology which we use in this paper, a *geographic feature* (or simply *feature*) is an abstraction of a real world phenomenon and can have various attributes that describe its *thematic* and *spatial* characteristics. For example, the country Greece is a feature, its name and population are thematic attributes, while its location on Earth in terms of polar co-ordinates is a spatial attribute. Knowledge about the spatial attributes of a feature can be *quantitative* or *qualitative*. For example, the fact that the distance between Athens and Salonika is 502 km is quantitative knowledge, while the fact that river Evros crosses Bulgaria and Turkey and is at the border of Greece with Turkey is qualitative knowledge. Quantitative geographic knowledge is usually represented using *geometries* (e.g., points, lines and polygons on the Cartesian plane) while qualitative geographic knowledge is captured by *qualitative binary relations* between the geometries of features.

A significant fraction of the available data on the Web is geospatial, and this fraction is growing by 20 percent or more per year [18]. Geospatial data can be qualitative and can be expressed as a property of an entity or an explicit assertion. For example, in the RDF dataset DBpedia[1] extracted from Wikipedia, resource `dbr:Berlin` has a data property `dbo:Country` with value `dbr:Germany` enabling the answering of questions such as “Cities in Germany” using DBpedia. Or a dataset like DBpedia can contain the fact `dbr:Berlin ogc:sfWithin dbr:Germany` and the question “Which cities are in Germany?” can again be answered using this dataset. Geospatial data can also be quantitative and can be expressed by a property of an entity which has value a geometry (latitude/longitude pair or polygon). Then, the question “Which cities are within 100 km of Berlin?” can be answered by retrieving the geometry of resource `dbr:Berlin` from an appropriate geospatial dataset, and then computing the distance of this geometry to the geometries of cities outside Berlin. In this paper, we focus on question answering from qualitative and quantitative geospatial knowledge made available on the Web as *linked open data.*

*This work has been funded by the Horizon 2020 Marie Sklodowska-Curie Innovative Training Network WDAqua (Answering Questions using Web Data) with grant agreement No. 642795.

GIR'18, November 6, 2018, Seattle, WA, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6034-0/18/11…$15.00
https://doi.org/10.1145/3281354.3281362

[1] http://wiki.dbpedia.org/

Examples of geospatial data published on the Web as linked open data include geospatial data from various countries (e.g., the United Kingdom[2] or The Netherlands[3]), OpenStreetMap data (published in RDF by project LinkedGeoData[4] [2] but also by our KR&R group at the National and Kapodistrian University of Athens[5]), and land cover/land use data sets (e.g., the European CORINE land cover dataset published in RDF by the same group in the context of various European projects). Queries over such data can be asked using the linked data query language SPARQL and its geospatial extensions GeoSPARQL[6] and stSPARQL [17]. However, to better serve the needs of non-technical end users, it would be worthwhile to offer a natural language interface to linked geospatial data based on QA techniques. To the best of our knowledge, none of the QA systems utilizing linked data developed in recent years [7] deals with geospatial data. The work presented in this paper makes the first steps towards this direction.

In a similar spirit, the GIR community has been emphasizing the need to develop techniques for answering geographic questions expressed in natural language over text data since 2004.[7] The importance of the research issues studied in GIR can also be seen by the fact that interaction with geospatial search engines on mobile devices today is often done using spoken natural language (e.g., in Google Maps you can ask for directions to a place and these directions are then spoken to you). This is in agreement with the vision of multimodal spatial querying presented in [27]. Geographical knowledge is also very important in the new generation of personal assistants such as Amazon Alexa or Google Home.

Important assets of the GIR and QA research communities are the *gold standards* i.e., datasets and sets of questions that can be used to test the effectiveness of developed systems and perform detailed comparisons of them. In the area of QA over linked data, such gold standards have recently been provided by the QALD challenge[8]. To the best of our knowledge, no gold standard for geospatial question answering over linked data has been proposed so far by QALD or any other relevant research activity.

The contributions of this paper are the following. We have designed and implemented GeoQA, the *first* question answering system for linked geospatial data. GeoQA is implemented using reusable components as part of the component-oriented Qanary question answering methodology [5, 29] and its most recent implementation Frankestein [30].

We have also developed a gold standard for question answering over linked geospatial data which consists of two parts. The first part is a linked geospatial dataset built from DBpedia, the GADM database of global administrative areas[9] and OpenStreetMap (OSM)[10]. For the purposes of the gold standard, GADM and OSM have been restricted to the United Kingdom and Ireland. The second part of the gold standard consists of 201 geospatial questions that have been collected by student volunteers at the National and Kapodistrian University of Athens. The gold standard is used in a preliminary evaluation of the effectiveness of GeoQA and it is also made freely available to the research community for evaluating other future proposals.[11] In this way, we contribute to a long-term research agenda towards question answering systems with geospatial features.

The rest of the paper is organized as follows. The next section presents related work. Section 3 presents the three datasets of the gold standard and the interlinking of GADM and OSM with DBpedia. Section 4 presents the gold standard questions. Section 5 presents our approach to building the query answering pipeline used by GeoQA. Section 6 presents a preliminary evaluation of GeoQA using the gold standard. Section 7 concludes the paper and discusses future work.

2 RELATED WORK

Since the first workshop in this field in 2004, question answering over textual data with geospatial information has been studied by Geographic Information Retrieval researchers. Relevant problems in this area include detecting place names (a special case of named entity recognition) and associated spatial natural language qualifiers in text and user queries, and disambiguating place names (a special case of named entity disambiguation). Two representative examples of systems where some of these issues have been studied are SPIRIT [16] and STEWARD [19].

An important evaluation initiative for geographic information retrieval from multilingual text has been GeoCLEF.[12] From the 2008 version of GeoCLEF, the GiKiP pilot is complementary to our paper since it concentrated on answering geospatial questions in three languages (Portuguese, English and German) from Wikipedia [26].

Query processing for linked geospatial data has been an active field of research recently culminating in the definition of the OGC standard GeoSPARQL, an extension of SPARQL with a vocabulary, datatypes and functions for expressing geospatial queries over linked data. There has also been substantial activity in the implementation of query processing systems such as Strabon [17] and Ontop-spatial [4], which both support GeoSPARQL. The work of the present paper goes beyond these query processors to offering question answering services over linked geospatial data, i.e. supporting queries expressed in natural language.

The work by Younis et al. [38] is most closely related to our work since it presents a system for answering geospatial questions over DBpedia. The system is based on a PostGIS[13] database containing precise geospatial information of features in the United Kingdom provided by Ordnance Survey, a spatial index of DBpedia resources built using their point coordinates, and a SPARQL endpoint storing the DBpedia dataset. The three classes of questions considered are proximity "Find churches within 1 km of the River Thames"), crossing (e.g., "Find the mouths of the rivers that cross Oxford") and containment (e.g., "Find churches in Manchester"). As we will see in section 5, these kinds of questions are a subset of the ones that can be handled by GeoQA. [38] informally discusses the techniques that can be used to answer such questions, but makes no system

[2] http://data.ordnancesurvey.co.uk/
[3] https://www.kadaster.nl/-/bag-linked-data
[4] http://linkedgeodata.org/About
[5] http://kr.di.uoa.gr/#datasets
[6] http://www.opengeospatial.org/standards/geosparql
[7] http://www.geo.uzh.ch/~rsp/gir18/
[8] https://qald.sebastianwalter.org/
[9] http://www.gadm.org/
[10] https://www.openstreetmap.org
[11] http://geoqa.di.uoa.gr/
[12] http://www.clef-initiative.eu/track/GeoCLEF
[13] http://postgis.net/

available with which we could compare our GeoQA. In addition, [38] makes no dataset available on which other approaches like ours can be evaluated. Finally, the discussion in the paper pays some attention to the quality of returned answers.

The paper [14] is also very relevant to our work. It explores the use of DBpedia and Geonames for answering topological queries involving administrative divisions of Switzerland and Scotland (since the authors are very familiar with the administrative geographies of these two countries). The paper contains a detailed discussion of quality issues in linked geospatial data and especially the two linked data sources used by the authors (e.g., incompleteness, inconsistency of data etc.). Finally, the paper considers queries for neighbouring and containing/contained administrative divisions, and measures precision and recall when only one of datasets or both linked datasets are used.

Another related paper is [33], which presents an inductive logic programming approach for learning a semantic parser and applies its techniques to two areas, one of which is querying geospatial databases. The authors of [33] have experimented with a dataset consisting of 1000 Prolog facts from the U.S. Geography domain, and have also developed a corpus of 880 natural language questions and their corresponding logical queries in Prolog.[14] A part of this corpus is used to train the semantic parser developed by the authors.

As we have already said, in the area of QA there is currently no engine that deals with geospatial questions like GeoQA. From the existing systems, PowerAqua needs to be mentioned in our context since it also assumes that questions will be answered from many datasets or ontologies [20].

3 CONSTRUCTING A GOLD STANDARD GEOSPATIAL DATA SET

In this section we discuss how to construct a gold standard geospatial dataset by interlinking DBpedia, OpenStreetMap and the GADM dataset of global administrative areas. Since DBpedia contains very limited geospatial information (e.g., latitude/longitude pairs, qualitative information via predicates such as `dbo:Country`), we enrich DBpedia with quantitative geospatial information (i.e., geometries) by interlinking it with OSM and GADM.

GADM is a dataset containing information about administrative divisions of various countries and their boundaries. The current version is 3.6 (released on 6 May 2018) and it contains information about 386,735 administrative areas. We have the data from previous release version 2.8(released on November 205) GADM gives the geometry of each administrative area as a multi-polygon and it also provides some other information such as its name and variant names. GADM has been available in linked data form since the publication of [25]. Since the dataset of [25] does not use the GeoSPARQL vocabulary or the newest version of GADM, we had to create it again from the available shapefiles using the tool GeoTriples[15]. For the purposes of this paper, we have only used GADM data from the United Kingdom (England, Scotland, Wales and Northern Ireland) and Ireland. The simple ontology we have used for GADM is shown graphically in[16] and it is also available publicly in RDF/XML format[17]. In the rest of the paper we use the prefix `gadmr:` instead of `http://www.app-lab.eu/gadm` for resources in the GADM data, and `gadmo:` for `http://www.app-lab.eu/gadm/ontology` for resources in the GADM ontology.

OSM is a collaborative project to create a free editable map of the world. It contains information about various features like rivers, lakes, cities, roads, points of interest (e.g., museums, restaurants and schools) etc. The geometries of these features can be points, lines or polygons. In addition to the geometry of a feature, OSM contains useful information such as name, feature class, layer etc. OSM data can be obtained in various formats. The first project to transform OSM data into RDF was LinkedGeoData [2]. Currently, this project does not provide an up-to-date version of OSM data that we could use for our study. For this reason, we had to repeat some of the work presented in [32] and, by doing this, go beyond [32] in the way that we will explain below. In the rest of the paper we use the prefix `osmr:` instead of `http://www.app-lab.eu/osm` for resources in the OSM data, and `osmo:` instead of `http://www.app-lab.eu/osm/ontology` for resources in the OSM ontology.

We obtained the OSM dataset in shapefile format from the company GEOFABRIK [18] and converted it into RDF using the GeoTriples tool. These shapefiles contains data available on date 30th August 2017. Like GADM, we have restricted our attention to the United Kingdom and Ireland. We designed a new ontology for OSM data which closely models the data in the shapefiles and made it publicly available in graphical format[19] and in RDF/XML format[20]. The ontology uses the GeoSPARQL vocabulary to model the geometries of various OSM features. Note that OSM does *not* have detailed administrative boundaries of various countries, this is why we rely on GADM for this information.

DBpedia is one of the most popular knowledge graphs derived from Wikipedia and its ontology which we use in the paper is publicly available[21]. In the rest of the paper we use the prefix dbo: instead of `http:dbpedia.org/ontology` for resources in the DBpedia ontology, and dbr: instead of `http://dbpedia.org/resource` for resources in the DBpedia knowledge graph. Interlinking of GADM and OSM with DBpedia allows us to answer geospatial questions that cannot be answered by any of the datasets in isolation. For example, the question"Which of the English counties that border Greater Manchester has the highest percentage of ethnic Asians?" can only be answered by consulting GADM to find the counties that border Greater Manchester, and then DBpedia to find the percentage of various ethnic groups in these counties. Also, the question "Which Greek politicians are graduates of a university located in a Greek island belonging to the region of Western Greece?" can be answered only by consulting all three datasets.

[14] http://www.cs.utexas.edu/users/ml/nldata/geoquery.html

[15] http://geotriples.di.uoa.gr

[16] http://geoqa.di.uoa.gr/images/gadm_ontology.png

[17] http://geoqa.di.uoa.gr/assets/GADM.owl

[18] http://download.geofabrik.de/europe.html

[19] http://sites.pyravlos.di.uoa.gr/dragonOSM.svg

[20] http://pyravlos-vm5.di.uoa.gr/osm.owl

[21] http://mappings.dbpedia.org/server/ontology/classes/

Table 1: Interlinking GADM with DBpedia

Country	Total entities	Linked automatically	Linked manually
UK	197	164	33
Ireland	27	17	10

3.1 Interlinking GADM with DBpedia

The interlinking of GADM with DBpedia was done as follows. We get the name of an administrative area from GADM and we query the DBpedia SPARQL endpoint to get the DBpedia resource having same label with the resource of GADM. Then we link these two resources using owl:sameAs. This simple procedure resulted in most of the GADM resources being linked. The remaining ones were linked manually. Table 1 gives the relevant numbers.

3.2 Interlinking of OSM with DBpedia

The task of interlinking OSM with DBpedia had some interesting challenges. First of all, we manually identified classes that have the same or very similar label in DBpedia and OSM. These classes are: Airport, Bank, Beach, Building, Canal, Castle, Cemetery, Church, City, College, Dam, Forest, Fort, Glacier, Golfcourse, Hospital, Hotel, Island, Library, Lighthouse, Locality, Memorial, Mine, Monument, Mosque, Museum, Park, Place, Prison, RailwayStation, Region, Restaurant, River, Road, School, Stadium, Stream, Temple, Theatre, Tower, Town, Tram, University, Village, Volcano, Watertower, Watermill, Windmill, and Zoo. Then, interlinking was done on a class-by class basis using the tool Silk. The OSM data is stored in a Strabon endpoint and the online DBpedia endpoint is used for the DBpedia data. The labels of the entities and the spatial distance of their geometries were considered equally for matching. In other words, we use the formula $(S(x, y) + MinDist(x, y))/2 = 1.0$ where

- x and y are the instances considered for matching in OSM and DBpedia respectively.
- $S(x, y)$ is the Levenshtein string similarity measure between the labels of x and y. The threshold taken for string similarity is 85%.
- $MinDist(x, y)$ is the minimum Euclidean distance between the geometries of x and y. After experimenting with different number of threshold values for Euclidean distance, we finalized the threshold to 1 kilometer.

Table 2 gives the number of instances of the various classes in both datasets, as well as the number of instances that were interlinked. The DBpedia instances have been selected by retrieving only the resources that have coordinates falling inside the minimum bounding rectangles of the geometries of the United Kingdom and Ireland. As it is expected, some classes have many more instances in one of the datasets. For example, the class Restaurant has 24055 instances in the subset of OSM that we consider and only 152 instances in DBpedia. Also, some of the classes having the same label are at different places in the class hierarchies of the two datasets. For example, the class Building is the parent class of Restaurant, Hotel, Hospital, Museum etc. in the DBpedia ontology, while it does not have any subclasses in the hierarchy of the OSM ontology, so we interlink instances of the subclasses of Building. Similarly, Road has subclasses that we consider in OSM ontology, while it does not have any subclasses in DBpedia. Naturally, when a class had zero instances in one or both datasets (e.g., Beach in the DBpedia subset we consider and Glacier in both datasets) then the class does not participate in the interlinking and does not appear in Table 2. Finally, we would like to mention that we found many misclassified instance in DBpedia in contrast to the other two datasets; this has also been pointed out in [32].

Let us now comment on some rows of Table 2 where there is an unexpectedly big difference in the number of instances in OSM and DBpedia for the same class. Let us take for example the class Airport. Unfortunately, the freely available OSM shapefiles for the United Kingdom and Ireland, provided by GEOFABRIK [22], contain only 7 airports (not even Heathrow airport of London is included!). On the contrary, DBpedia has a rather large number resources classified as airports. In some cases, these are wrongly classified e.g., dbr:Brahan_Castle, a castle, is wrongly classified as dbo:Airport. It is also interesting to consider the row for class River. There are many more instances of River in OSM than in DBpedia because OSM has a different entry for each of the segments/polygons making up a river in its full length. The same issue exists for the classes Canal and Stream. This is the reason that the number of total interlinked instances is bigger than the cardinality of the intersection of the two datasets for classes like River. There are also other classes where the difference in instances between OSM and DBpedia is very big due to the nature of the knowledge in the datasets. For example, DBpedia has information about only 1339 hotels in its full dataset of which 212 hotels are in the United Kingdom and Ireland. The corresponding number of hotels in OSM is 9819 hotels as we see from the table. In a similar way, the class Restaurant in DBpedia has few instances compared to OSM.

After completing the interlinking with Silk, there were some entities that were not linked. These were checked and linked, if appropriate, manually. For matches below 100 all matching pairs were checked manually for correctness. For larger numbers of matching pairs, we checked manually 100 random pairs and found them all to be correct. So, we conclude that our matching process is very accurate.

Comparing the interlinking of OSM and DBpedia that we have done with the interlinking done in LinkedGeoData [32], we can see that we have interlinked instances belonging to many more classes. The OSM dataset in the case of LinkedGeoData is stored using Virtuoso which has support only for point geometries. Therefore, no queries involving complex geometries can be done, and the interlinked resources in the case of LinkedGeoData are limited to OSM nodes.

The GADM and OSM datasets as well as the interlinking dataset is publicly available on the Web site of the gold standard.[23] We will call this data part of the gold standard *GeoData201*.

[22] http://download.geofabrik.de/europe.html

[23] http://geoqa.di.uoa.gr

Table 2: Interlinking OSM with DBpedia

Class label	No. of Instances in OSM	No. of Instances in DBpedia	Inter-linked Instances	Inter-linked Instances (semi auto-matically)	Total Inter-linked Inst-ances
Airport	7	815	1	5	6
Bank	7621	29	1	2	3
Canal	7902	167	2171	920	3091
Castle	1357	486	161	36	197
City	86	101	45	18	63
College	1529	38	0	2	2
Dam	330	26	1	3	4
Hospital	2352	537	244	149	393
Hotel	9819	212	73	81	154
Island	2477	750	219	138	357
Library	3635	119	47	25	72
Light-house	423	39	9	14	23
Monu-ment	2108	38	5	3	8
Museum	2313	933	327	219	546
Park	54830	382	252	103	355
Prison	207	199	28	119	137
Railway-Station	3932	45	0	0	0
Region	13	151	0	0	0
Rest-aurant	24058	152	31	30	61
River	52897	785	4342	237	4579
School	33217	5556	2683	691	3374
Stadium	799	687	120	78	198
Stream	240293	470	885	265	1150
Theatre	1224	86	19	33	52
Tower	2373	35	0	0	0
Town	1960	1066	132	18	150
Univer-sity	2466	1099	169	41	210
Village	15743	15346	4308	4087	8395

4 CREATING A GOLD STANDARD SET OF GEOSPATIAL QUESTIONS

To be able to evaluate the effectiveness of our query engine and compare it with other QA engines available, we have created a new benchmark set of 201 questions which we have collectively called *GeoQuestions201*. The questions have been written by third-year students of the 2017-2018 Artificial Intelligence course in our department. The students were asked to target the above three data sources by imagining scenarios where geospatial information will be needed and could be provided by an intelligent assistant, and to propose questions with a geospatial dimension that they considered "simple" (a few examples of such questions were provided). The authors of the paper have then "cleaned" the given set of questions and produced the SPARQL or GeoSPARQL queries that correspond to them assuming ontologies that describe the three data sources using the GeoSPARQL vocabulary. The complete set of resources (data sources, ontologies, natural language questions and SPARQL/GeoSPARQL queries) are available on the Web at `http://geoqa.di.uoa.gr`.

The questions in the benchmark GeoQuestions201 fall under the following categories:

(1) *Asking for the location of a feature.* For example, "Where is Loch Goil located?". In GeoQA, these questions can be answered by posing a SPARQL query to DBpedia. Google can also answer such questions precisely.

(2) *Asking whether a feature is in a geospatial relation with another feature.* For example, "Is Liverpool east of Ireland?". The geospatial relation in this example question is a cardinal direction one (east of). Other geospatial relations in the set of questions include topological (borders) or distance (near or "at most 2km from"). In GeoQA, these questions are answered most of the time by using GADM and OpenStreetMap because the relevant qualitative geospatial knowledge is not present in DBpedia and/or the detailed geometries of features are needed for evaluating the geospatial relation of the question. Google cannot answer such factoid questions; it can only return a list of relevant Web pages.

(3) *Asking for features of a given class that are in a geospatial relation with another feature.* For example, "Which counties border county Lincolnshire?" or "Which hotels in Belfast are at most 2km from George Best Belfast City Airport?". The geospatial relation in the first example question is a topological one (border). As in the previous category, other geospatial relations in the set of questions include cardinal (e.g., southeast of) or distance (near or "at most 2km from" as in the second example question). In GeoQA, these questions can be answered by using not just DBpedia but also GADM and OpenStreetMap when the detailed geometries of features are needed for evaluating the geospatial relations. Google can also answer such questions precisely in many but not all cases (e.g., it can answer precisely the first and third questions but not the second).
Questions in this category might also have a second geospatial relation and a third feature which are used to further constrain the second feature. For example, "Which restaurants are near Big Ben in London?" or "Which rivers cross London in Ontario?". In the first question, we have also provided some more information about Big Ben although this might not have been necessary.[24] In the second question, "in Ontario" is used to make clear that we are referring to the city London in Ontario, Canada not the more well-known city of London in England.[25]

(4) *Asking for features of a given class that are in a geospatial relation with any features of another class.* For example, "Which churches are near castles?". Arguably, this category of questions might not be useful unless one specifies a geographical area of interest; this is done by the next category of questions.

[24] The authors of this paper are not aware of another Big Ben.

[25] Boringly enough, London, Ontario is also crossed by a Thames river. We bet this is not how this river was called by native Indians in 1534 when Canada was discovered.

(5) *Asking for features of a given class that are in a geospatial relation with an unspecified feature of another class which, in turn, is in another geospatial relation with a feature specified explicitly.* An example of such a question is "Which churches are near a castle in Scotland?". Google cannot answer such questions precisely.

(6) *The questions in this category are like the ones in Categories 3 to 5 above, but in addition, the thematic and/or geospatial characteristics of the features that are expected as answers (i.e., the features of the first class mentioned in the question) satisfy some further condition (e.g., numeric).* For example, "Which mountains in Scotland have height more than 1000 meters?" or "Which villages in Scotland have a population of less than 500 people?" or "Is there a church in the county of Greater Manchester dedicated to St. Patrick?" or "Which Greek restaurants in London are near Wembley stadium?". In these examples, the extra attribute conditions may require GeoQA to consult all three data sources to find the answer to a question. Google can answer precisely the first, third and fourth example question, but not the second, since its knowledge graph does not contain population information for villages in Scotland.

(7) *Questions with quantities and aggregates.* For example, "Which is the highest mountain in Ireland?" or "Which hotel is the nearest to Old Trafford Stadium in Manchester?" or "Which is the largest lake by area in Great Britain?" Questions with aggregates and quantities are currently considered an open area of research in question answering [37] and are the current emphasis of GeoQA. Interestingly, Google can answer all three example questions precisely.
Note that questions in this class might also exhibit features of the previous two classes e.g., when a topological relation is involved or when the condition on an attribute refers to a quantity (e.g., height of a mountain). Such questions cannot be handled by QA engines and Google at the moment. For example, the question "Which is the largest county of England by population which borders Lincolnshire?" is answered incorrectly by Google (county Bristol is given as the answer). The list of benchmark questions is available publicly[26].

5 CREATING A GEOSPATIAL QUESTION ANSWERING PIPELINE

We now present our approach to translating a natural language question into a GeoSPARQL query that can be executed on the union of the datasets presented in the previous section. For this, we build a geospatial question answering system using Qanary [8] and Frankenstein [30].

5.1 The Frankenstein Framework for Building QA Systems

Qanary is a lightweight component-based QA methodology for the rapid engineering of QA pipelines [5, 6]. Frankenstein [30] is the most recent implementation of the ideas of Qanary; this makes it an excellent framework for developing reusable QA components and integrating them in QA pipelines. Frankenstein is built using the formal methodology of [5] and uses standard RDF technology to wrap and integrate existing standalone implementations of state-of-the-art components that can be useful in a QA system. The Qanary methodology is driven by the knowledge available for describing the input question and related concepts during the QA process. Frankenstein uses an extensible and flexible vocabulary [29] for data exchange between the different QA components. This vocabulary establishes an abstraction layer for the communication of QA components. While integrating components using Frankenstein, all the knowledge associated with a question and the QA process is stored in a process-independent knowledge base using the vocabulary. Each component is implemented as an independent micro-service implementing the same RESTful interface. During the start-up phase of a QA pipeline, a service registry is automatically called by all components. As all components are following the same service interface and are registered to a central mediator, they easily can be activated and combined by developers to create different QA systems.

5.2 GeoQA: A Geospatial QA System

In our work, we leverage the power of the Frankestein framework to create six QA components which collectively implement the geospatial QA pipeline of GeoQA. The QA process of GeoQA uses the following modules implemented as components in the Frankenstein framework: dependency parse tree generator, concept identifier, instance identifier, geospatial relation identifier, SPARQL/-GeoSPARQL query generator and SPARQL/GeoSPARQL query executor. Our components are fully integrated in the Frankenstein ecosystem and can be reused to implement geospatial features in other QA systems, as our implementation is not monolithic like the implementation of many other QA systems [10, 21, 34].

GeoQA takes as input a question in natural language (currently only English is supported) and the three linked geospatial datasets presented in Section 3, and produces one or more answers that are resources of the given datasets. Question answering is performed by translating the input question to a set of SPARQL or GeoSPARQL queries, ranking these queries, and executing the top ranked query over two endpoints using the SPARQL `SERVICE` keyword. For DBpedia, we use its public Virtuoso endpoint[27] while for GADM, OSM and their interlinking dataset we use a Strabon endpoint. In Figure 1, we present the conceptual view of the implemented GeoQA system. The various components of GeoQA are discussed below.

Dependency parse tree generator. This component carries out part-of-speech tagging and generates a dependency parse tree for the input question using the Stanford CoreNLP software. The dependency parse tree is produced in CoNLL-U format [23].

Concept identifier. The concept identifier module identifies the *types of features* specified by the user in the input question and maps them to the corresponding classes in the DBpedia, GADM and OSM ontologies. We use the equivalent ontology-oriented term *concept* for a feature type in this paper. For example, if the input question is "Which restaurants are near Big Ben in London?", then the

[26] http://geoqa.di.uoa.gr/benchmarkquestions.html

[27] http://dbpedia.org/sparql

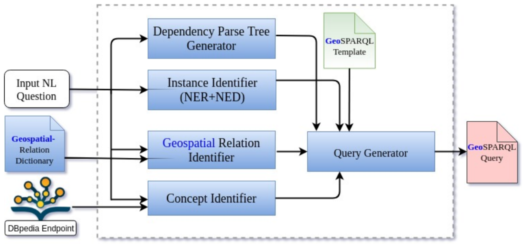
Input NL Question
Geospatial-Relation Dictionary
DBpedia Endpoint
Dependency Parse Tree Generator
Instance Identifier (NER+NED)
Geospatial Relation Identifier
Concept Identifier
GeoSPARQL Template
Query Generator
GeoSPARQL Query

Figure 1: The conceptual architecture of the GeoQA system

Table 3: Geospatial relation categories and relations

Category	Geospatial relation
Topological relations	“within”, “crosses”, “borders”
Distance relations	“near”, “at most x units”, “at least x units”
Cardinal direction relations	“north of”, “south of”, “east of”, “west of”, “northwest of”, “northeast of”, “southwest of”, and “southeast of”

term “restaurants” is identified as a feature type and mapped to the class `osmo:Restaurant` in the OSM ontology and `dbo:Restaurant` in the DBpedia ontology. The matching classes are found using string matching on the labels of the classes (the Java library function `java.util.regex.Pattern.matcher()`is used) together with lemmatization from Stanford CoreNLP and synonyms from Wordnet. In its final stage, the concept identifier annotates the appropriate node of the dependency parse tree with its results.

Instance identifier. The next useful information to be identified in an input question is the *features* mentioned. These can be e.g., the country Ireland or the city Dublin or the river Shannon etc. We use the equivalent ontology-oriented term *instance(s)* for features in this paper. Once instances are identified, they are mapped to DBpedia, OSM or GADM resources. For this functionality, the existing components of named entity recognition (NER) and named entity disambiguation (NED) of Frankenstein are used in sequence. Stanford NER [12] implements the NER task and AGDISTIS [35] implements the NED task for DBpedia. We also search for resources in the OSM and GADM dataset that have the same label as the entity identified by Stanford NER, and add them to the list of identified instances. For illustration, consider the input question “Which airports are in London?”. The term “London” is the identified instance (feature) by Stanford NER, and it is disambiguated to the DBpedia resource `dbr:London` by AGDISTIS, and to `osmr:england/places/id/107775` and `gadmr:administrativeUnit_GBR_adm2_56` by our code. In its final stage, the instance identifier annotates the appropriate node of the dependency parse tree with its results.

Geospatial relation identifier. Geospatial questions such as the ones targeted by GeoQA almost always include a qualitative geospatial relation such as “borders” or a quantitative one such as “at most 2km from”. The current implementation supports the 14 geospatial relations shown on Table 3. These include some topological, some distance and some cardinal direction relations [11, 13, 31]. Table 4 gives a dictionary of the various synonyms for these relations that can appear instead of them in a question. The semantics of topological relations is as in the dimensionally extended 9-intersection model []. Qualitative spatial relations of proximity like “close to", “near" etc. are translated into (rather arbitrary) quantitative distance relations based on the concept identified earlier by the concept identifier (e.g., when asking for “hotels near a place", “near” is taken to mean at most 1 kilometer). The semantics of cardinal direction relations are the usual ones i.e., a relation A north of B is given meaning by considering the bounding box of the reference region B and the partition of the plane in nine areas that is induced by it [31]. The same semantics are implemented by the Strabon system and its query language stSPARQL which is used as our back end geospatial RDF store. [17] GeoSPARQL does not support any cardinal direction functions or relations. Finally, the paper [36] gives a more intuitive semantics of cardinal directions for polygons, but an implementation of this semantics is more expensive computationally than the semantics used in Strabon.

Like the previous modules, this module first identifies geospatial relations in the input question, and then maps them to a spatial function of the GeoSPARQL or stSPARQL vocabulary, or a data property with a spatial semantics in the DBpedia ontology. As we have already discussed in the introduction, DBpedia contains limited explicit or implicit geospatial knowledge using latitude/longitude pairs, and properties such as `dbp:northeast` for cardinal direction relations or class-specific properties such as `dbo:city` (e.g., for class `dbr:River`). GeoQA does not make use of quantitative geospatial information (i.e., latitude/longitude pairs) from DBpedia since we have more detailed geospatial knowledge in the form of polygons in the datasets GADM and OSM. However, it does makes use of qualitative geospatial knowledge from DBpedia expressed using the data properties just mentioned (although this knowledge is rather scarce as discussed in [24]). As an example, for the question “Which counties border Lincolnshire?”, the geospatial relation “borders” is identified from the verbs in the dependency tree, and it is mapped to the spatial function `geof:sfTouches` of the GeoSPARQL vocabulary.

In its final stage, the geospatial relation identifier annotates the appropriate node of the dependency parse tree with its results. In the near future, GeoQA will cover all the prototypical spatial relations shown experimentally to correspond to natural language utterances by Egenhofer, Mark and their colleagues in [9, 22, 28].

Some previous question answering engines such as QUINT [1] jointly disambiguate utterances in a question by expressing the relevant problem as a constrained optimization problem and utilizing state-of-the-art Integer Linear Programming solvers for solving it efficiently. We plan to also consider this approach in the future.

Query generator. This module creates a SPARQL or a GeoSPARQL query using handcrafted query templates. From gathering questions from Google Trends and also studying the questions in our gold standard, we have identified the question patterns shown on Table 5. In this table C stands for “concept”, I for “instance” and R for “geospatial relation” following the terminology we have introduced above. For each pattern, the table gives an example question

Table 4: Geospatial relations and their synonyms

Geospatial relation	Synonyms in dictionary
within	in, inside, is located in, is included in
crosses	cross, intersect
near	nearby, close to, around
borders	is/are at the border of, is/are at the outskirts of, at the boundary of
north of	above of
south of	below
east of	to the right
west of	to the left

Listing 1: SPARQL/GeoSPARQL Query for Motivating Example

```
SPARQL:
select ?x
where {
    ?x rdf:type dbo:River.
    ?x dbo:city dbr:Limerick.
}

GeoSPARQL:
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX osmo: <http://www.app-lab.eu/osm/ontology#>

select ?x
where {
    ?x rdf:type osmo:River;
       geo:hasGeometry ?xGeom.
       ?xGeom geo:asWKT ?xWKT.

    gadmr:Limerick geo:hasGeometry ?iGeom.
    ?iGeom geo:asWKT ?iWKT.

    FILTER(geof:sfCrosses(?xWKT, ?iWKT))
}
```

Listing 2: GeoSPARQL Query for Motivating Example

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX osmo: <http://www.app-lab.eu/osm/ontology#>
select ?x
where {
        ?x rdf:type osmo:Forest;
           geo:hasGeometry ?xGeom.
        ?xGeom geo:asWKT ?xWKT.
        ?instance owl:sameAs dbr:Manchester;
                   geo:hasGeometry ?iGeom.
        ?iGeom geo:asWKT ?iWKT.
        FILTER(geof:distance(?xWKT,?iWKT,uom:metre) <= 5000)
        }
```

and the corresponding GeoSPARQL and/or SPARQL query template. The query templates contain slots (strings starting with an underscore) that can only be identified when an example question is encountered and will be completed by the query generator (see below).

Table 5: Supported question patterns with examples and corresponding SPARQL/GeoSPARQL query templates

Pattern	Example natural language question	Templates
CRI	Which rivers cross Limerick?	SPARQL: select ?x where { ?x rdf:type _Concept. ?x _Relation _Instance. } GeoSPARQL v1: select ?x where { ?x rdf:type _Concept; geo:hasGeometry ?xGeom. ?xGeom geo:asWKT ?xWKT. _Instance geo:hasGeometry ?iGeom. ?iGeom geo:asWKT ?iWKT. FILTER(_Relation(?xWKT, ?iWKT)) } GeoSPARQL v2: select ?x where { ?x rdf:type _Concept; geo:hasGeometry ?xGeom. ?xGeom geo:asWKT ?xWKT. ?instance owl:sameAs _Instance; geo:hasGeometry ?iGeom. ?iGeom geo:asWKT ?iWKT. FILTER(_Relation(?xWKT, ?iWKT)) }
CRIRI	Which churches are close to the Shannon in Limerick?	select ?x where { ?x rdf:type _Concept; geo:hasGeometry ?xGeom. ?xGeom geo:asWKT ?xWKT. _Instance1 geo:hasGeometry ?i1Geom. ?i1Geom geo:asWKT ?i1WKT. _Instance2 geo:hasGeometry ?i2Geom. ?i2Geom geo:asWKT ?i2WKT. FILTER(_Relation1(?xWKT, ?i1WKT) && _Relation2(?i1WKT, ?i2WKT)) }

CRC	Which restaurants are near hotels?	select ?x where {
		?x rdf:type _Concept1;
		geo:hasGeometry ?xGeom.
		?xGeom geo:asWKT ?xWKT.
		?y rdf:type _Concept2;
		geo:hasGeometry ?yGeom.
		?yGeom geo:asWKT ?yWKT.
		FILTER(_Relation(?xWKT, ?yWKT)) }

CRCRI	Which restaurants are near hotels in Limerick?	select ?x where { ?x rdf:type _Concept1; geo:hasGeometry ?xGeom. ?xGeom geo:asWKT ?xWKT. ?y rdf:type _Concept2; geo:hasGeometry ?yGeom. ?yGeom geo:asWKT ?yWKT. _Instance geo:hasGeometry ?zGeom. ?zGeom geo:asWKT ?zWKT. FILTER(_Relation1(?xWKT, ?yWKT) && _Relation2(?xWKT, ?zWKT) && _Relation2(?yWKT, ?zWKT) }
IRI	Is Hampshire north of Berkshire?	ASK where { _Instance1 geo:hasGeometry ?iGeom1. ?iGeom1 geo:asWKT ?iWKT1. _Instance2 geo:hasGeometry ?iGeom2. ?iGeom2 geo:asWKT ?iWKT2. FILTER(_Relation(?iWKT1, ?iWKT2)) }

For each input question, the slots in the template are replaced by the query executor with the output of the previous modules, to generate a SPARQL or a GeoSPARQL query. For example, for the question “Which rivers cross Limerick?”, the identified pattern is CRI. The question pattern is identified by searching the dependency parse tree in which nodes have been annotated with the results of the concept, instance and geospatial relation identifier modules presented above. If the question does not follow any of the patterns, a message is passed to the next component that no query has been generated. The appropriate templates are selected from Table 5, their slots are filled with the resources identified earlier and the corresponding GeoSPARQL or SPARQL queries are generated. Here the concepts are `dbr:River` from DBpedia and `osmo:River` from OSM, the instances are `gadmr:Limerick` from GADM and `osmo:irelandandnorthernireland/places/id/2518952` from OSM, and the geospatial relations are `dbo:city` from the DBpedia ontology and the GeoSPARQL function `geof:sfCrosses`.

The row of Table 5 for pattern CRI contains two GeoSPARQL queries (v1 and v2). The second query is for the case when the identified instance is a DBpedia resource for which geometry information is available in GADM or OSM. This is where the `owl:sameAs` sentences produced by our interlinking process discussed in Section 3 are used. Listing 2 shows an example of this case for the question “Which forest is near Manchester?”. Similar templates exist for all the other patterns, but are not shown in Table 5 due to space considerations.

The last job of the query generator is to rank the generated queries. Query ranking is a crucial component of a QA. In the current version of GeoQA, we use a very simple heuristic for the ranking of generated queries based on the component of GeoData201 that is used for obtaining the geospatial knowledge used in the query. We rank queries using DBpedia higher than the ones using GADM which are, in turn, ranked higher than the ones using OSM. We break ties randomly. The idea behind this is to generate the simplest query that will be evaluated very efficiently by the RDF store Strabon. Arguably, one might not prefer the simple query, but rather the one giving us the best precision and recall even though it might be costlier to evaluate.

Another way to increase recall is to use more than one component of GeoData201 for answering a question. For example, for the question “Which towns in England are east of Manchester?” DBpedia gives us 3 answers (Glossop, Stallybridge and Hyde) while OSM gives us 1626 towns. Another way to increase precision is to have the query generator take into account more *schema information* from the ontologies of the three datasets. This is illustrated by the SPARQL query in Listing 1 where we make use of the fact that the property `dbo:city` is used in DBpedia to refer to the cities crossed by a river. We plan to have all such information in the query generator for each pair of class and geospatial relation and for each of the three datasets. This approach has also been taken in [38]. Another promising state-of-the-art approach for ranking is *learning to rank* by using e.g., a random forest classifier [3]. We are also currently investigating this approach for use in GeoQA by using the publicly available module of QUINT[28]. We are actively experimenting with such improvements to the query generator.

Table 6: Evaluation of GeoQA

Gold Questions	Answered Questions	Precision	Recall	F1
86	44	37.38%	41.43%	35.50%

Expressive Power of Patterns. It is interesting to consider the expressive power of patterns in Table 5 by giving a corresponding binary first-order logic formula.[29] Questions following the CRI pattern can be written formally as $x : C(x) \wedge (\exists i)R(x, i)$. Questions following the pattern CRIRI can be written as $x : C(x) \wedge (\exists i_1)(\exists i_2)(R_1(x, i_1) \wedge R_2(i_1, i_2))$. Questions following the pattern CRC can be written as $x : C_1(x) \wedge (\exists i)(C_2(i) \wedge R(x, i))$. Finally, questions written as CRCRI can be written as $x : C_1(x) \wedge (\exists i_1)(\exists i_2)(R(x, i_1) \wedge C_2(i_2) \wedge R_2(i_1, i_2))$.

Query executor. The last module executes the top-ranked SPARQL or GeoSPARQL query against a Strabon [17] endpoint which also communicates with a DBpedia endpoint through the use of the `SERVICE` keyword in queries. If no query has been generated, the user is notified that the question could not be answered.

6 PRELIMINARY EVALUATION

The current version of the GeoQA engine presented above has been preliminarily evaluated using the gold standard dataset GeoData201 and questions GeoQuestions201 presented in Sections 3 and 4. GeoQA was run using the 86 questions that fall under Categories 2 to 5. It is not difficult to see from the discussion of the GeoQA pipeline of Section 5.2 that the current version of GeoQA is targeted to these four classes of questions.

Table 6 summarizes the effectiveness of GeoQA on the gold standard using the well-known metrics of precision, recall and F-measure. We calculate precision, recall and F-measure for the 86 questions. For the 29 questions (out of 86), GeoQA does not identify the correct question pattern and as a result no query is generated. From the 57 generated queries, only 44 queries return results. The remaining 13 queries do not generate any answers, although such answers exist in the GeoData201 dataset, because of errors of the instance identifier component. The instance identifier depends on the entity recognizer Stanford NER as discussed above. For the 42 questions that either cannot be answered or we return an empty set of answers incorrectly, Table 7 shows, for each specific question, which module fails to generate proper results. As an example, for the question Q6 “Which hotels are near Big Ben?”, Stanford NER identifies “Ben” as a named entity which is then disambiguated by AGDISTIS to resource `dbr:The_Bens` of `rdf:type dbo:Band` in DBpedia, and from our code to OSM resource `osmr:irelandandnorthernireland/places/id/4203174` of `rdf:type osmo:locality`. According to our ranking heuristic, the OSM resource is then used in the generated query which fails to return any answers.

[28] https://github.com/abujabal/learning-to-rank

[29] In the following formulas, we assume that identifiers (i.e., geographic features) are denoted by constants, concepts (i.e., classes of features) by unary predicates and geospatial relations by binary predicates. Constants and predicates are denoted by capital letters while variables are denoted by lowercase letters. Variables are assumed to range over identifiers. The “:” symbol should be read as “such that”.

Table 7: Questions with no answers

Module Responsible	Question Number
NER	Q56,Q269,Q333,Q17,Q56,Q92,Q98,Q283, Q312,Q320
NED	Q201,Q272,Q273,Q335,Q114,Q241,Q245
Concept Identifier	Q123,Q205,Q323,Q333,Q335,Q117
Relation Identifier	Q65,Q120,Q129,Q236,Q268,Q290,Q323
Query Generator	Q61,Q71,Q119,Q182,Q220,Q235,Q242, Q304,Q319,Q330,Q106,Q237

7 CONCLUSIONS

In this paper we have addressed the challenges of providing access to linked geospatial data for non-expert users using natural language QA interfaces. Given the use of geospatial contexts in many practical situations this challenge is of major importance while adopting QA for wide use. Our main contribution was the implementation of GeoQA which is, to the best of our knowledge, the first QA engine which is able to answer questions with a geospatial dimension. We have also evaluated GeoQA using a gold standard dataset and set of questions.

In future work, we plan to work on the following topics. We will refine the current algorithm of GeoQA so that we improve the effectiveness of various components, and in this way increase the disappointing precision, recall and F1 that we have now. We have already identified deficiencies of some of the components and discussed them above. We will carry out a more detailed evaluation of GeoQA to account for cases where the geospatial information taken from DBpedia needs to be augmented with geospatial information from GADM and OSM to increase the recall of the algorithm. This is also the approach taken in [38]. We will also concentrate on more complex questions, especially questions including conditions (e.g., “Which rivers cross Limerick and their length is more than 300km?”) and questions involving aggregates and quantities (e.g., “Which is the biggest county by area in England?”). Finally, we will consider text (e.g., travel blogs etc.) as another rich source of geographic knowledge and make GeoQA able to discover and exploit such sources using techniques from GIR.

REFERENCES

[1] A. Abujabal, M. Yahya, M. Riedewald, and G. Weikum. [n. d.]. Automated Template Generation for Question Answering over Knowledge Graphs. In *International Conference on World Wide Web, year = 2017*.

[2] S. Auer, J. Lehmann, and S. Hellmann. 2009. LinkedGeoData: Adding a Spatial Dimension to the Web of Data. In *ISWC*.

[3] Hannah Bast and Elmar Haussmann. 2015. More Accurate Question Answering on Freebase. In *International Conference on Information and Knowledge Management, CIKM*.

[4] K. Bereta and M. Koubarakis. 2016. Ontop of Geospatial Databases. In *ISWC*.

[5] A. Both, D. Diefenbach, K. Singh, S. Shekarpour, D. Cherix, and C. Lange. 2016. Qanary - A Methodology for Vocabulary-Driven Open Question Answering Systems. In *Extended Semantic Web Conference*.

[6] A. Both, K. Singh, D. Diefenbach, and I. Lytra. 2017. Rapid Engineering of QA Systems Using the Light-Weight Qanary Architecture. In *Web Engineering - International Conference*.

[7] D. Diefenbach, V. López, K. Deep Singh, and P. Maret. 2018. Core techniques of question answering systems over knowledge bases: a survey. *Knowl. Inf. Syst.* 55, 3 (2018).

[8] D. Diefenbach, K. Singh, A. Both, D. Cherix, C. Lange, and S. Auer. 2017. The Qanary Ecosystem: Getting New Insights by Composing Question Answering Pipelines. In *Web Engineering - International Conference*.

[9] M. P. Dube and M. J. Egenhofer. 2012. An Ordering of Convex Topological Relations. In *GIScience*.

[10] M. Dubey, S. Dasgupta, A. Sharma, K. Höffner, and J. Lehmann. 2016. AskNow: A Framework for Natural Language Query Formalization in SPARQL. In *Extended Semantic Web Conference*.

[11] Max J. Egenhofer and Robert D. Franzosa. 1991. Point Set Topological Relations. *International Journal of Geographical Information Systems* 5, 2 (1991).

[12] J. Rose Finkel, T. Grenager, and C. D. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Annual Meeting of the Association for Computational Linguistics*.

[13] A. U. Frank. 1992. Qualitative spatial reasoning about distances and directions in geographic space. *J. Vis. Lang. Comput.* 3, 4 (1992).

[14] R. Grütter, R. S. Purves, and L. Wotruba. 2017. Evaluating Topological Queries in Linked Data Using DBpedia and GeoNames in Switzerland and Scotland. *Trans. GIS* 21, 1 (2017).

[15] L. Hirschman and R. J. Gaizauskas. 2001. Natural language question answering: the view from here. *Natural Language Engineering* 7, 4 (2001).

[16] C. B. Jones, R. Purves, A. Ruas, M. Sanderson, M. Sester, M. J. van Kreveld, and R. Weibel. 2002. Spatial information retrieval and geographical ontologies an overview of the SPIRIT project. In *ACM SIGIR*.

[17] K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis. 2012. Strabon: A Semantic Geospatial DBMS. In *ISWC*.

[18] J.-G. Lee and M. Kang. 2015. Geospatial Big Data: Challenges and Opportunities. *Big Data Research* 2, 2 (2015).

[19] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. 2007. STEWARD: architecture of a spatio-textual search engine. In *ACM SIGSPATIAL*.

[20] V. López, M. Fernández, E. Motta, and N. Stieler. 2012. PowerAqua: Supporting users in querying and exploring the Semantic Web. *Semantic Web* 3, 3 (2012).

[21] V. López, E. Motta, and V. S. Uren. 2006. PowerAqua: Fishing the Semantic Web. In *Extended Semantic Web Conference*.

[22] D. M Mark and M. J Egenhofer. 1995. Topology of prototypical spatial relations between lines and regions in English and Spanish. In *AUTOCARTO*.

[23] J. Nivre, M.-Catherine de Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. T. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty, and D. Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *International Conference on Language Resources and Evaluation, LREC*.

[24] B. Regalia, K. Janowicz, and S. Gao. [n. d.]. VOLT: A Provenance-Producing, Transparent SPARQL Proxy for the On-Demand Computation of Linked Data and its Application to Spatiotemporally Dependent Data. In *Extended Semantic Web Conference, year = 2016*.

[25] J Salas and A. Harth. 2011. Finding spatial equivalences accross multiple RDF datasets. In *Terra Cognita workshop*.

[26] D. Santos, N. Cardoso, P. Carvalho, I. Dornescu, S. Hartrumpf, J. Leveling, and Y. Skalban. 2008. Getting Geographical Answers from Wikipedia: the GikiP pilot at CLEF. In *CLEF*.

[27] I. Schlaisich and M. J. Egenhofer. 2001. Multimodal spatial querying: what people sketch and talk about. In *HCI International*.

[28] A. Rashid B. M. Shariff, M. J. Egenhofer, and D. M. Mark. 1998. Natural-Language Spatial Relations Between Linear and Areal Objects: The Topology and Metric of English-Language Terms. *International Journal of Geographical Information Science* 12, 3 (1998).

[29] K. Singh, A. Both, D. Diefenbach, S. Shekarpour, D. Cherix, and C. Lange. 2016. Qanary - The Fast Track to Creating a Question Answering System with Linked Data Technology. In *Extended Semantic Web Conference*.

[30] K. Singh, A. Sethupat Radhakrishna, A. Both, S. Shekarpour, I. Lytra, R. Usbeck, A. Vyas, A. Khikmatullaev, D. Punjani, C. Lange, M. E. Vidal, J. Lehmann, and S. Auer. 2018. Why Reinvent the Wheel: Let's Build Question Answering Systems Together. In *International Conference on World Wide Web, WWW*.

[31] S. Skiadopoulos and M. Koubarakis. 2001. Composing Cardinal Direction Relations. In *Advances in Spatial and Temporal Databases, SSTD*.

[32] C. Stadler, J. Lehmann, K. Höffner, and S. Auer. 2012. LinkedGeoData: A core for a web of spatial open data. *Semantic Web* 3, 4 (2012).

[33] L. R. Tang and R. J. Mooney. 2001. Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing. In *EMCL*.

[34] C. Unger, L. Bühmann, J. Lehmann, A-C. Ngonga Ngomo, D. Gerber, and P. Cimiano. 2012. Template-based question answering over RDF data. In *International Conference on World Wide Web*.

[35] R. Usbeck, A. Ngonga Ngomo, M. Röder, D. Gerber, S. Athaide Coelho, S. Auer, and A. Both. 2014. AGDISTIS - Agnostic Disambiguation of Named Entities Using Linked Open Data. In *European Conference on Artificial Intelligence*.

[36] Marc J. van Kreveld and Iris Reinbacher. 2004. Good news: Partitioning a simple polygon by compass directions. *Int. J. Comput. Geometry Appl.* 14, 4-5 (2004).

[37] G. Weikum. 2018. Beyond Knowledge Graphs: New Frontiers of Machine Knowledge. Invited talk at the Wiki Workshop. Held at The Web Conference 2018 in Lyon, France.

[38] E. M. G. Younis, C. B. Jones, V. Tanasescu, and A. I. Abdelmoty. 2012. Hybrid Geo-spatial Query Methods on the Semantic Web with a Spatially-Enhanced Index of DBpedia. In *GIScience*.