Developing geospatial web applications using question answering engines, knowledge graphs and linked data tools*

Sergios-Anestis Kefalidis[1], Konstantinos Plas[1], George Stamoulis[1], Dharmen Punjani[2], Pierre Maret[2] and Manolis Koubarakis[1,3](✉)

[1] National and Kapodistrian University of Athens, Athens, Greece
{s.kefalidis,kplas,gstam,koubarak}@di.uoa.gr
[2] Laboratory Hubert Curien, Université St. Monnet, St. Etienne, France
dharmen.punjani@gmail.com, pierre.maret@univ-st-etienne.fr
[3] Archimedes/Athena RC, Marousi, Greece

Abstract. We show how to develop geospatial web applications using the geospatial question answering engine GeoQA2, the geospatial knowledge graph YAGO2geo, the spatiotemporal RDF store Strabon and the Web-GIS tool Sextant. We demonstrate the combined functionality of these tools by developing PnyQA, a system for exploring data from the 2020 presidential election of the United States using natural language questions.

Keywords: geospatial data · question answering · knowledge graph · spatiotemporal RDF store · web visualization

1 Introduction

A *knowledge graph (KG)* is a directed graph where nodes represent entities (e.g., the football team Olympiacos Piraeus or the person José Luis Mendilibar) and edges represent relationships between entities (e.g., that Mendilibar is the coach of Olympiacos) or attributes of entities and their values (e.g., that Olympiacos was founded in 1925). KGs are typically encoded as sets of *triples* in the RDF data model and queried using the RDF query language SPARQL. Examples of well-known KGs are DBpedia [2], Wikidata [28] and YAGO [27].

A *geospatial KG* is a KG where nodes represent geographic features (e.g., the country Greece) and edges represent relationships between features (e.g., the capital of Greece is Athens or Bulgaria is north of Greece) or attributes of features and their values. Attributes can represent *thematic* information (e.g., Greece has population 10 million) or *geospatial* information (e.g., the geometry of

* This work has received funding from the project STELAR (101070122), under the European Union's Horizon Europe research and innovation programme. This work has also been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

Greece is the multipolygon "..." in the coordinate reference system WGS84). To the best of our knowledge, five geospatial KGs are available, each with a different emphasis (YAGO2 [9], YAGO2geo [12], WorldKG [6], KnowWhereGraph [11] and the KG of Böckling et al. [4]). Geospatial KGs can also be expressed as RDF triples and can be queried by SPARQL or more appropriately with GeoSPARQL.

YAGO2geo (`https://yago2geo.di.uoa.gr/`) is a KG that has been developed by our group by extending the KG YAGO2 [9] with detailed geospatial information about: (i) administrative divisions data of all countries using the GADM dataset (`https://gadm.org/`), (ii) official administrative divisions data for the countries of Greece, United Kingdom, Ireland and United States, and (ii) some categories of OpenStreetMap features (e.g., natural features like water bodies and man-made features such as cities). YAGO2 models geographic space by defining *geoentities* with point geometries consisting of latitude/longitude pairs by utilizing data from gazeteer GeoNames [9]. YAGO2geo contains all the geoentities of YAGO2 and extends them with more detailed geometries such as lines, polygons and multipolygons taken from the sources mentioned above. Finally, YAGO2geo includes *new* geoentities present in the above administrative datasets and OpenStreetMap that were not present in YAGO2. YAGO2geo currently contains 703 thousand polygons and 3.8 million lines.

Question answering (QA) over KGs is the research area which studies how to answer questions expressed in natural language over KGs (e.g., "Which team won the UEFA Conference League in the season 2023-2024?"). In this paper we are interested in using QA engines that answer *geospatial* questions (e.g., "Which countries border Greece to the north?" or "Which river flows through London?" or "What is the largest lake by area in Scotland?") over geospatial KGs.

GeoQA [25] and its revised version [26] was the first geospatial QA engine to be developed by our team (AI Team, `https://ai.di.uoa.gr/`). GeoQA can answer geospatial questions over the KG DBpedia interlinked with the parts of GADM and OpenStreetMap for the United Kingdom and Ireland.

GeoQA has recently been re-engineered into version 2 [24] and it is available as open source[4]. The new version targets the union of the KG YAGO2 and the geospatial KG YAGO2geo, and it improves GeoQA by having been optimized in various ways and being able to answer a greater variety of questions. In [14] our group presented the dataset GeoQuestions1089 which contains 1089 triples of geospatial questions, their answers, and the respective SPARQL/GeoSPARQL queries. GeoQuestions1089 is currently the largest geospatial QA benchmark and it is made freely available to the research community by our group[5]. GeoQuestions1089 contains simple questions like "Which countries border Greece?", as well as, semantically complex questions that require a sophisticated understanding of both natural language and GeoSPARQL in order to be answered (e.g., "Is the total size of lakes in Greece larger than lake Loch Lomond in Scotland?". [14] uses GeoQuestions1089 to evaluate the effective-

[4] `https://github.com/AI-team-UoA/GeoQA2`
[5] `https://github.com/AI-team-UoA/GeoQuestions1089`

ness of GeoQA2 and its competitor engine developed by Hamzei et al. [8] and shows that GeoQA2 performs better.

Using only YAGO2 and YAGO2geo as the data sources of GeoQA2 may limit its usefulness in applications. Even though answering geospatial questions over a geospatial KG is a reasonable use case, in the real world, geospatial data from a KG is typically used together with other kinds of thematic data to produce useful results. This has been our experience in many European projects such as TELEIOS and ExtremeEarth (`https://earthanalytics.eu/`) where we used geospatial KGs together with data extracted from satellite images to develop web applications e.g., for wildfire monitoring [17] and smart farming [1]. The KG that GeoQA2 targets (the union of YAGO2 and YAGO2geo, which will refer to as *YAGO2+geo* for conciseness) has a useful but limited amount of thematic and geospatial information. Therefore, to unlock the real potential of GeoQA2, one needs to infuse the KG with *application-specific thematic data* as well and, in this way, *a powerful tool for analyzing thematic data with a geospatial dimension is born.*

In this demo paper, we present PnyQA[6], a system that realizes the approach sketched in the previous paragraph and, using GeoQA2 as its core, enables a user to analyze the results of the 2020 U.S. Presidential Election on the county level, using natural language questions. Our demo is similar in spirit with the demo [19] which shows how to explore the geospatial KG KnowWhereGraph using faceted search. The distinguishing feature of our work is that of having natural language as a means of expressing user requests, a functionality which is complementary to the search functionalities of the demo of [19].

2 System Architecture

Figure 1 shows the architecture of PnyQA. In the following, we describe its main software components and overall functionality.

GeoQA2. GeoQA2 takes as input a question in English and the YAGO2+geo KG, and produces a set of answers. QA is performed by translating the input question into a SPARQL or GeoSPARQL query, which is subsequently executed over an RDF store (Strabon or other) endpoint that contains the target KGs.

We present the GeoQA2 pipeline which contains the following main components: dependency parse tree generator, concept identifier, instance identifier, geospatial relation identifier, property identifier, query generator and query transpiler. The functionality of these components will be discussed below using the question "Is the largest island in the United Kingdom larger than Crete by population?".

The *dependency parse tree generator* carries out part-of-speech (POS) tagging and generates a dependency parse tree for the input question using the Stanford CoreNLP toolkit [20].

[6] The name was inspired by the hill Pnyka, where the Ancient Athenians held their assemblies and public votes.

The *concept identifier* identifies the *types of features (concepts)* present in the input question (e.g., "island") and maps them to the corresponding classes of the target KG ontology (e.g., `y2geoo:OSM_island`). These concepts are identified and mapped to the ontology classes of YAGO2 and YAGO2geo using n-gram string similarity.

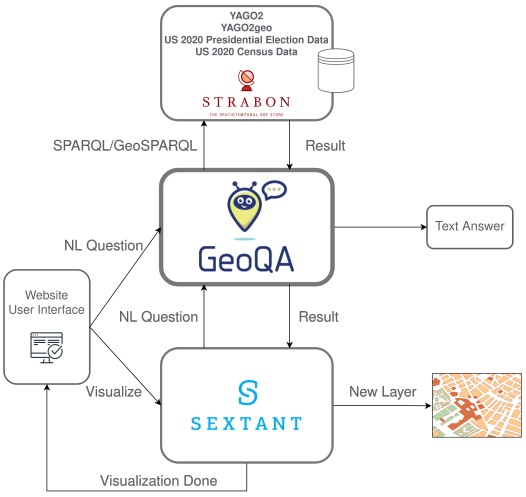

Fig. 1. The conceptual architecture of PnyQA

The *instance identifier* identifies the *features* (*instances*) present in the input question (e.g., "United Kingdom" and "Crete"). Then, these elements are mapped to KG resources (e.g., `yago:United_Kingdom` and `yago:Crete`) following a two-step approach. First, the TagMeDisambiguate tool [7] is used to identify and link named entities to Wikipedia pages. Subsequently, the KG entity that best matches the Wikipedia page is located in the KG. In [13] we have evaluated 12 tools with similar functionality on GeoQuestions1089, including the well-known ones ELQ [18] and GENRE [5], and found that TagMeDisambiguate has the best accuracy for our task.

The *geospatial relation identifier* identifies the geospatial relations (e.g., "in") in the input question and maps them to the respective spatial function of the GeoSPARQL or stSPARQL vocabulary (e.g., `geof:sfWithin`) according to a mapping between geospatial relations and stSPARQL/GeoSPARQL functions provided by a dictionary.

The *property identifier* identifies *attributes of features or types of features* specified by the user in input questions and maps them to the corresponding properties in the target KG. For instance, for the example question the property "population" of type of feature "island" will be identified and mapped to property `yago:hasPopulation`. The attributes in the input question are identified based

on the POS tags NN, JJ, NNP and NP generated by the dependency parsing process and the concepts/instances identified by earlier steps.

The *query generator* produces the GeoSPARQL query corresponding to the input question by using a set of query templates, heuristics and the annotated parse tree. For questions containing aggregates and superlatives (e.g., "largest"), the query generator constructs also the constituency parse tree of the input question and uses it to modify the templates to support these kinds of questions.

The *query transpiler* is responsible for rewriting the generated query to benefit from offline geospatial optimizations, speeding up the query in the process. The transpiler is also available as a standalone application on our team's GitHub page[7].

In addition to GeoQA2, our demo uses two linked geospatial data tools developed by our team (see [15] for a recent overview of our work in this area).

Strabon. Strabon is a state-of-the-art spatiotemporal RDF store that supports storing spatiotemporal RDF data and evaluating spatiotemporal queries on it, using the query languages stSPARQL [16] and the OGC standard GeoSPARQL. According to benchmarks Geographica 2 [10], Strabon is one of the most powerful, in terms of performance and functionality, geospatial RDF stores. Our system is built in a way that supports multiple RDF stores, Strabon could be replaced by any other RDF-store that is capable of executing GeoSPARQL and stSPARQL queries.

Sextant. Sextant is a Web-GIS tool that enables users to browse and visualize geospatial data in formats such as KML, GML, and TIFF files. It also facilitates communication with SPARQL endpoints, allowing the generation of map layers based on the results obtained from GeoSPARQL or stSPARQL queries. This feature allows Sextant to visualize the results of the GeoSPARQL queries generated by GeoQA as layers over the map. Sextant has been under continuous development at the National and Kapodistrian University of Athens since its initial publication [21]. It was and it still is the most functional tool in the literature for visualizing linked geospatial data.

System Architecture. The aforementioned components are combined to model the architecture shown in Figure 1. The Website UI facilitates the input of questions in natural language by the user and offers the option to select the desired output format, which consists of textual answers or visual representation of the obtained results. The natural language questions are processed by a modified version of GeoQA which produces a SPARQL/GeoSPARL query that is executed in Strabon, which in turn returns the results to GeoQA. Results in text form are directly returned through GeoQA. If the user requests the visualization of the results, the query results are passed to Sextant to be visualized according to the user's choice. Sextant further provides the capability to visualize multiple questions simultaneously into a single map, as separate layers.

[7] `https://github.com/AI-team-UoA/GoST`

Availability In addition to the individual components of our system which have already been released to the public, PnyQA is made publically available as a service through an endpoint hosted on our group's server infrastructure at `https://pnyqa.di.uoa.gr`.

Performance Optimization. As we alluded to previously, our system makes use of offline optimizations to handle the large number of geometric calculations that it has to perform, which often leads to very long response times. For instance, checking whether a geometry is within a large administrative area with complex borders is computationally a very challenging task. Hence, to improve the time performance, we *pre-computed and materialized* certain relations between entities in the YAGO2geo knowledge graph that change infrequently.

During the evaluation of GeoQA2 [14], we observed that the topological geospatial relations "within", "crosses", "intersects", "touches", "overlaps", "covers" and "equals" require expensive computations, while "near", "north", "south", "east" and "west" are easily computed. Hence, we decided to materialize the above costly topological relations. The aforementioned transpiler rewrites queries that contain these topological relations from GeoSPARQL to SPARQL. This optimization is particularly beneficial, since, as shown in [14], it greatly boosts the performance of computationally demanding queries.

One of the major concerns related to materialization is the size of resulting knowledge, and the overhead that this can cause to its processing. Overall, the materialized version of YAGO2geo had 17,210,176 more triples, which in terms of system memory, amounts to about 3GB and 10.21% increased in total size, but as shown in [14], it does not affect the performance of the question answering system negatively. The time required to calculate the implied geospatial relation was close to 5 days, which can be considered negligible, as it happens offline, and it is being repeated infrequently (only when the knowledge graph changes). The calculation of the implied relations was facilitated by utilizing a distributed implementation of the algorithm GIA.nt [22], implemented in the system DS-JedAI [23] [8].

3 Presentation and Demonstration

In our presentation, we will introduce the architecture of our system and showcase its functionality with two scenarios about analyzing the results of the 2020 U.S. Presidential Election. These scenarios showcase the two different workflows of PnyQA. Our analysis will include a blend of geospatial and demographic information on the U.S. county level. To enable that, we use the following datasets:

– For geospatial information, we use the KG YAGO2+geo. As was discussed previously, YAGO2+geo is the primary target of GeoQA2.

[8] `https://github.com/GiorgosMandi/DS-JedAI`

- County-level election results for the 2020 U.S. Presidential Election[9], which are made available by the MIT Election Data and Science Lab. This dataset includes the vote count per political party for each county in the United States. Depending on the state that a county belongs to, it can also include information about the voting methods used, e.g., ballot voting.
- General population demographic data from the 2020 U.S. Census[10], as they are made available by the U.S. Census Bureau. This dataset contains, among other things, statistics about age, gender and race on a county level.

The two thematic datasets are linked to YAGO2+geo and loaded in an RDF store. Linking is done by executing a script that introduces information of the two thematic datasets as new RDF triples in YAGO2+geo. The resource files of GeoQA2 are updated accordingly to correspond to the new ontology.

The first workflow of the demonstration does not involve the visualization capabilities of Sextant since the results are in text form, and is the following:

1. The user inputs the question "Which counties south of Kansas were won by the Democratic Party?" and selects the text-answer workflow.
2. The question is passed to GeoQA2 which translates it to a GeoSPARQL query (Listing 1.1).
3. GeoQA2 requests the execution of the generated query from the RDF store endpoint, which in turn returns a list of counties.
4. The list of counties is displayed on the Web Interface.

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX yago: <http://yago-knowledge.org/resource/>
PREFIX y2geoo: <http://kr.di.uoa.gr/yago2geo/ontology/>
PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
PREFIX usel: <http://kr.di.uoa.gr/us/election/ontology/>

SELECT ?county WHERE
{
    yago:Kansas geo:hasGeometry ?geom . ?geom geo:asWKT ?mWKT .
    ?county y2geoo:hasGADM_Description \"County\" .
    ?county geo:hasGeometry ?geom2 .
    ?geom2 geo:asWKT ?cWKT .
    FILTER (strdf:above(?mWKT, ?cWKT)) .
    ?county usel:hasElection_Data ?data .
    ?data usel:hasDemocrat_Vote ?dv .
    ?data usel:hasRepublican_Vote ?rv .
    FILTER (?dv > ?rv) .
}
```

Listing 1.1. GeoQA2-generated GeoSPARQL query

The second workflow of our demonstration utilizes all components of our system and the end result is a multi-layered map:

1. The user inputs the question "In which counties of Texas with less than 30,000 inhabitants did the Democratic Party get more votes than the Republican Party?" and selects the visualization/Sextant workflow.

[9] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/VOQCHQ

[10] https://data.census.gov/table?g=010XX00US,$0500000&d=DEC+Demographic+Profile&tid=DECENNIALDP2020.DP1

2. Since the visualization workflow was selected, the input question is interpreted as a request for the creation of a layer on a map and the question is passed to GeoQA2.
3. GeoQA2 translates the question to a GeoSPARQL query.
4. Identically to the first workflow, GeoQA2 requests the execution of the generated query from the RDF store endpoint, which in turn returns a list of county geometries.
5. GeoQA2 returns the results to Sextant which takes care of the visualization (shown in Figure 2).

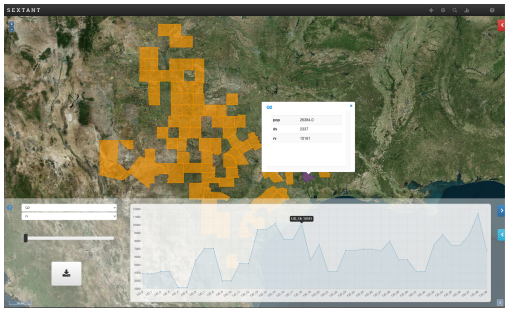

Fig. 2. Sextant map for workflow 2

At this point, it is important to note that the user can continue inputting questions, the results of which are shown as additional layers on the map. For example, continuing the previous example:

6. The user inputs a question, e.g., "In which counties of Georgia with less than 30,000 inhabitants did the Democratic Party get less votes than the Republican Party?". The Sextant workflow is already selected.
7. Steps (2) - (5) are repeated. The visualization is updated with a second layer.
8. When the user is finished, he presses the "Finish Visualization" button and the workflow is complete.

For our presentation, in addition to showcasing these two workflows, we will present and showcase the PnyQA pipeline on the aforementioned scenarios, along with the functionality of each pipeline component. Additionally, we will encourage WISE 2024 participants to ask their own questions in natural language to further illustrate our system and demonstrate the stages of our architecture through their remarks. For each question that we will demonstrate or that a participant will experiment with, we will also show what modern chatbots such as ChatGPT, Gemini and Claude answer when given the same question. Our experiments on August 15, 2024 for the above three questions show that Gemini and Claude cannot answer such questions (i.e., they have not been trained

on relevant election data) while ChatGPT gives informative but incomplete answers. In [13] we have compared the accuracy of GeoQA2 with ChatGPT on the benchmark dataset GeoQuestions1089 and found a similar situation. The problem of developing large language models for geospatial data is an open research problem where interesting research is carried out currently (see e.g., [3]).

4 Outlook

In current work we are developing GeoQA3, a more robust and effective QA engine which uses large language models in various stages of the question answering pipeline (e.g., the instance identifier and the query generator).

References

1. Appel, F. et al.: ExtremeEarth: Managing water availability for crops using earth observation and machine learning. In: EDBT (2023)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: A nucleus for a web of open data. In: ISWC ASWC (2007)
3. Balsebre, P., Huang, W., Cong, G.: LAMP: A language model on the map. CoRR **abs/2403.09059** (2024)
4. Böckling, M., Paulheim, H., Detzler, S.: A planet scale spatial-temporal knowledge graph based on OpenStreetMap and H3 grid. CoRR **abs/2405.15375** (2024)
5. Cao, N.D., Izacard, G., Riedel, S., Petroni, F.: Autoregressive entity retrieval. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), `https://openreview.net/forum?id=5k8F6UU39V`
6. Dsouza, A., Tempelmeier, N., Yu, R., Gottschalk, S., Demidova, E.: WorldKG: A world-scale geographic knowledge graph. In: CIKM (2021)
7. Ferragina, P., Scaiella, U.: TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In: CIKM (2010)
8. Hamzei, E., Tomko, M., Winter, S.: Translating place-related questions to geosparql queries. In: WWW (2022)
9. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. Artif. Intell. **194** (2013)
10. Ioannidis, T., Garbis, G., Kyzirakos, K., Bereta, K., Koubarakis, M.: Evaluating geospatial RDF stores using the benchmark Geographica 2. J. Data Semant. **10** (2021)
11. Janowicz, K. et al.: Know, Know Where, Knowwheregraph: A Densely Connected, Cross-Domain Knowledge Graph and Geo-Enrichment Service Stack for Applications in Environmental Intelligence. AI Mag. **43**(1), 30–39 (2022)
12. Karalis, N., Mandilaras, G.M., Koubarakis, M.: Extending the YAGO2 knowledge graph with precise geospatial knowledge. In: ISWC (2019)
13. Kefalidis, S., Punjani, D., Tsalapati, E., Plas, K., Pollali, M.A., Maret, P., Koubarakis, M.: The question answering system geoqa2 and a new benchmark for its evaluation. International Journal of Applied Earth Observation and Geoinformation **134**, 104203 (2024). `https://doi.org/https://doi.org/10.1016/j.jag.2024.104203`, `https://www.sciencedirect.com/science/article/pii/S1569843224005594`

14. Kefalidis, S., Punjani, D., Tsalapati, E., Plas, K., Pollali, M., Mitsios, M., Tsokanaridou, M., Koubarakis, M., Maret, P.: Benchmarking geospatial question answering engines using the dataset GeoQuestions1089. In: ISWC (2023)
15. Koubarakis, M. (ed.): Geospatial Data Science: A Hands-on Approach for Building Geospatial Applications Using Linked Data Technologies, vol. 51. Association for Computing Machinery, New York, NY, USA, 1 edn. (2023)
16. Koubarakis, M., Kyzirakos, K.: Modeling and querying metadata in the semantic sensor web: The model stRDF and the query language stSPARQL. In: ESWC (2010)
17. Kyzirakos, K. et al.: Wildfire monitoring using satellite images, ontologies and linked geospatial data. J. Web Semant. **24** (2014)
18. Li, B.Z., Min, S., Iyer, S., Mehdad, Y., Yih, W.: Efficient one-pass end-to-end entity linking for questions. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020. pp. 6433–6441. Association for Computational Linguistics (2020). `https://doi.org/10.18653/V1/2020.EMNLP-MAIN.522`, `https://doi.org/10.18653/v1/2020.emnlp-main.522`
19. Liu, Z. et al.: Knowledge explorer: exploring the 12-billion-statement knowwhere-graph using faceted search (demo paper). In: SIGSPATIAL (2022)
20. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations. pp. 55–60. The Association for Computer Linguistics (2014). `https://doi.org/10.3115/V1/P14-5010`, `https://doi.org/10.3115/v1/p14-5010`
21. Nikolaou, C., Dogani, K., Bereta, K., Garbis, G., Karpathiotakis, M., Kyzirakos, K., Koubarakis, M.: Sextant: Visualizing time-evolving linked geospatial data. J. Web Semant. **35**, 35–52 (2015)
22. Papadakis, G., Mandilaras, G.M., Mamoulis, N., Koubarakis, M.: Progressive, holistic geospatial interlinking. ACM / IW3C2 (2021)
23. Papamichalopoulos, M., Papadakis, G., Mandilaras, G., Siampou, M.D., Mamoulis, N., Koubarakis, M.: Three-dimensional geospatial interlinking with jedai-spatial (2022)
24. Punjani, D., Kefalidis, S., Plas, K., Tsalapati, E., Koubarakis, M., Maret, P.: The question answering system GeoQA2. In: GeoKG & GeoAI (2023)
25. Punjani, D. et al.: Template-based question answering over linked geospatial data. In: Proceedings of the 12th Workshop on Geographic Information Retrieval (2018)
26. Punjani, D. et al.: Template-based question answering over linked geospatial data. CoRR **abs/2007.07060** (2020)
27. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: WWW (2007)
28. Vrandecic, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)