

The EOBS Knowledge Graph: A Knowledge Graph of European Climate Observations

Ammar Yousaf, Auriol Degbelo*

Chair of Geoinformatics, TU Dresden, Germany

Abstract

The increasing availability of long-term climate datasets has opened new opportunities for data-driven analysis. Yet, these datasets are often distributed in formats such as NetCDF that remain difficult to query, integrate, and reuse beyond specialized workflows. The E-OBS gridded observational dataset (1950–2024), for example, provides valuable climate information across Europe, yet its use typically requires domain expertise and custom scripting, limiting accessibility for broader communities. To mitigate this issue, we present EOBS-KG, a knowledge graph (3.2 billion triples) that provides E-OBS data as a semantically structured and queryable resource. Observations are represented as instances linked to geo-referenced grid cells, with measurement results encoded as structured quantities and organized into observation collections. We demonstrate the utility of the knowledge graph through the EOBS-Agent, an LLM-based agent that enables natural language querying of the EOBS-KG via a structured planning–query–execution–verbalization workflow. We release 1) the EOBS-KG, 2) the Python-based pipeline for the knowledge graph generation, and 3) the code of the EOBS-Agent, both as a script and as a Docker container.

Keywords

climate data, geographic knowledge graph, scientific knowledge graph, LLM agent

1. Introduction

The increasing availability of long-term climate datasets has opened new opportunities for data-driven analysis. Yet, these datasets are often distributed in formats such as NetCDF that remain difficult to query, integrate, and reuse beyond specialized workflows. Concurrently, large language models (LLMs) have opened up new opportunities for information retrieval. There are several opportunities for fruitful work at the intersection of LLMs and knowledge graphs. The most relevant for the current article is that knowledge graphs can provide context to LLMs for retrieval-augmented generation, helping LLMs (partially) mitigate their well-known difficulties with numerical values [1]. Applications such as Climate Bot [2], ChatClimate [3], and ClimateChat [4] exist for general-purpose question answering (QA) in the context of the climate domain. However, the primary source of information for these applications has been web documents [4], reports [2, 3, 4], and academic papers [2, 3, 4]. The use of real-world datasets as information sources has, so far, received little attention, and the knowledge graph introduced in this work intends to address this gap. Through the enrichment of raw data with semantic annotations, it provides a starting point for factoid QA and analytical QA (and hence novel interfaces that facilitate the collection of evidence about the climate state in Europe). In line with Mai et al [5], factoid QA addresses questions whose answers are facts that can be looked up in a knowledge base; analytical QA addresses questions that require analytical workflows (i.e. chaining of computations) to arrive at an answer.

The work done in this article is in line with the argument that smart data opens up novel application scenarios [6]. More concretely, a knowledge graph of climate observations provides structured, semantically-rich data about climate observations, and hence facilitates scientific data reuse by machines. This comes with three benefits. First, the knowledge graph can be used as a benchmarking source while testing new approaches for (analytical) geographic question answering in the Semantic Web. Second, the dataset can provide an additional source that (LLM) agents can use for fact-checking, mitigating the often-quoted hallucination problem. Third, a knowledge graph of climate observations -

Posters, Demos, Blue Sky, and Tutorials at SEMANTICS 2026, Sep 2026, Ghent, Belgium

*Corresponding author.

✉ ammar.yousaf@tu-dresden.de (A. Yousaf); auriol.degbelo@tu-dresden.de (A. Degbelo)

ORCID 0000-0001-5087-8776 (A. Degbelo)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

combined with an LLM agent - enables basic information extraction through natural language, making the information in these scientific datasets accessible to non-experts, in addition to trained professionals. This is particularly important for climate observations, since currently the interaction with these datasets happens primarily through interactive visualizations (for examples see [7, 8]), scripting and programming interfaces (e.g. [9]). The long-term vision of this work is a knowledge graph of interconnected (climate) observations about the planet that can be leveraged to gather and synthesize evidence about changes in the environment. As a starting point, the E-OBS dataset [10, 11] is taken as an example. The main contributions of the work are 1) EOBS-KG, a knowledge graph of numerical facts from 1950 to 2024 about climate observations in Europe, and 2) an Agent that uses that knowledge graph to facilitate natural language question answering about climate observations in Europe.

2. The EOBS Knowledge Graph

2.1. Data source

The EOBS-KG is built upon the E-OBS dataset [10, 11] (version 31.0), a widely used gridded observational climate dataset covering Europe. The E-OBS dataset is produced within the European Climate Assessment & Dataset (ECA&D) initiative. E-OBS combines station-based measurements with interpolation techniques to generate spatially continuous climate observations on a regular latitude–longitude grid, making it suitable for large-scale and long-term climate analysis. The data are distributed in NetCDF [12] (Network Common Data Form) format, which is commonly used for storing multidimensional scientific data. In this format, observations are organized along three main dimensions: time, latitude, and longitude. In this work, we use the dataset at a spatial resolution of 0.1° , which provides detailed coverage across the European region. The complete temporal range of the dataset, from 1950 to 2024, is included, and all available variables in the archive were considered. The focus on the machine-interpretable representation of numerical observations for a specific region is what sets the current work apart from previous climate-related knowledge graphs and ontologies, for instance: the Climate Change Knowledge Graph [13] (focus: model-based climate projections), LinkClimate [14] (focus: integration of datasets from different sources), and [15] (focus: provenance assessment).

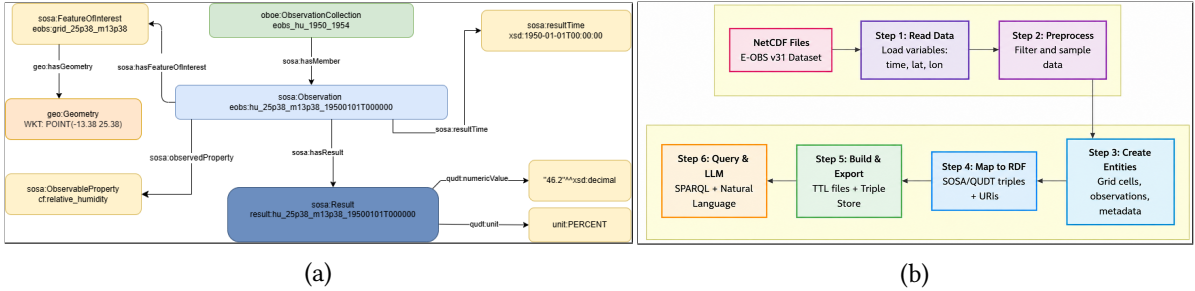


Figure 1: (a) E-OBS Data Model. (b) Transformation pipeline from NetCDF to RDF.

2.2. Transformation: Ontologies and Data Model

We reuse several well-known ontologies to ensure semantic interoperability (see Figure 1a). The SOSA ontology [16] provides the core structure for modeling observations, sensors, observable properties, and results. Measurement values are represented using the QUDT ontology [17], enabling standardized representation of quantities and units. Spatial information is modeled using the GeoSPARQL Ontology [18] and coordinates are represented using Well-known Text (WKT) [19]. Observable properties are aligned with the Climate and Forecast (CF) standard name vocabulary [20], ensuring consistency with domain-specific semantics in climate science. The transformation pipeline (Figure 1b) makes use of the Extensible Observation Ontology (OBOE) [21] to represent observation collections. Standard RDF, RDFS, and XML Schema (XSD) vocabularies are used for typing, labeling, and datatype definitions.

The knowledge graph adopts a consistent URI scheme to uniquely identify all resources. As part of the transformation pipeline, URIs are generated programmatically, primarily using variable names, spatial coordinates, and timestamps. Observations are identified using the pattern `eobs:{var}_{lat}_{lon}_{timestamp}`, where latitude and longitude values are formatted to two decimal places and the decimal point is replaced by the character `p`, to avoid invalid characters (e.g. decimal points) in URIs. Grid cells, representing spatial features of interest, are defined using the pattern `grid:{lat}_{lon}`. The naming for observation collections follows the pattern `eobs:eobs_{var}_{start_year}_{end_year}`, where each resource represents a collection of observations grouped by a specific climate variable and temporal range. The transformation pipeline is implemented in Python and converts NetCDF data into RDF triples through a sequence of processing steps. The pipeline was executed on a high-performance computing (HPC) environment using parallel processing across multiple compute nodes to handle the large volume of climate data. The complete transformation of the E-OBS dataset resulted in approximately 3.2 billion RDF triples and required continuous execution over a period of approximately three days (72 hours). The use of distributed computation was essential to achieve this scale of transformation within a practical timeframe.

2.3. Resource: The EOBS Knowledge Graph

The graph includes five observable properties (`precipitation_amount`, `surface_downwelling_shortwave_flux_in_air`, `air_temperature`, `relative_humidity`, and `wind_speed`) and contains 3,275,869,149 RDF triples about climate observations across Europe (Table 1). The graph includes 363,964,049 instances of `sosa:Observation`, each representing a spatio-temporal climate measurement. These observations are linked to 4,416 distinct grid cells, which represent fixed spatial locations across Europe. The graph further includes 99 instances of `oboe:ObservationCollection`, grouping observations by variable and temporal range.

Accessibility. To facilitate reuse, the data is made available in multiple forms. The dataset is provided as RDF dumps in Turtle format, organized into temporal batches corresponding to five-year intervals. In addition, the knowledge graph is accessible through a SPARQL endpoint. The transformation pipeline, supporting scripts, and configuration files are also shared to facilitate reuse and further development.

Licensing. As mentioned above, the knowledge graph is derived from the E-OBS dataset, which is provided through the European Climate Assessment & Dataset (ECA&D). The E-OBS dataset’s terms of use stipulate that they “are strictly for use in non-commercial research and education projects only” [11]. Hence, the EOBS Knowledge Graph is released under CC BY-NC-SA 4.0 (Attribution-NonCommercial-ShareAlike 4.0 International) in line with the original restrictions.

Table 1
Summary statistics about the knowledge graph.

Metric	Value	Number of observation/variables	Value
Total RDF triples generated	3,275,869,149	Precipitation	63,804,831
Total processing time	~72 hours	Mean Temperature	188,908,067
Number of observations	363,964,049	Relative Humidity	45,645,936
Number of grid cells (lat/lon as geo:Geometry)	4,416	Radiation (Shortwave Flux)	38,523,987
Number of observation collections	99	Wind speed	27,081,228

3. Evaluation

We demonstrate the value of the knowledge graph through two use cases. First, we implement and test an LLM-based agent for question answering over European climate observations, showing thereby the usefulness of the knowledge graph for automated (geographic) question answering. Second, we

show how the knowledge graph can be leveraged for federated search through examples of queries that combine information from the EOBS knowledge graph with that from external sources (Wikidata).

3.1. Usage 1: LLM Agent

The EOBS-Agent enables natural language access to the EOBS Knowledge Graph through a LangGraph-based architecture. The system integrates a web-based frontend (see Figure 2) with a backend workflow that orchestrates query planning and query execution, and answer verbalization. The interaction with the frontend and the backend happens through an API. Query interpretation happens through a template-based design, that is, the agent selects an appropriate SPARQL query template from a predefined set. This template-driven design (as opposed to approaches such as SPARQLGen [22], Chain-of-thought Prompting [23], or in-context-learning [24], aiming at general solutions to the Text-to-SPARQL translation problem) ensures controlled query generation and reduces the risk of producing invalid or unsupported queries for a specific use case. To test the EOBS-Agent, we use a set of competency questions, following existing strategies for ontology evaluation [25, 26]. At this stage, the evaluation is theory-based; that is, the evaluation questions were designed based on the (visual) information-seeking mantra [27]: overview-first, then zoom-and-filter, then details-on-demand. As discussed in [27], gaining an overview of the entire collection, zooming in on items of interest or filtering out uninteresting items, and selecting an item or group to get details when needed, are core tasks during data exploration. The current evaluation set contains 16 question types and 48 questions. A test case was created for each question type, and each test case includes three semantically equivalent variations of a natural-language question, an expected SPARQL template, expected coverage items, a category, and a difficulty label. Examples of question types are shown in Table 2. The evaluation assessed both average execution time and success rate for the natural language queries, averaged over five runs. For each run, a question was randomly selected from the three available formulations for a test case. Two different approaches were used to estimate the success rate: LLM-as-a-Judge [28] (i.e. compare bindings and questions to see if the answer has likely been successful) and manual assessment. The model used for the agent is gpt-4o-mini, while the model used to evaluate the results is gpt-5.4-nano. The average execution time was 30.71s (range: [1.38s – 61.11s]). The average accuracy rate was [LLM-as-a-Judge] 80% (range: [68.8% - 87.5%]) and [manual inspection] 90% (range: [81.2% - 100%]). The reason for these discrepancies is that the LLM-Judge often marked missing information as ‘errors’ if the SPARQL query lacked an explicit projection of a relevant variable (e.g. “The query is intended to provide an overview of climate observations for 2024, but the data returned does not specify that it is for the year 2024”). The most common causes of failures include difficulty interpreting queries lacking keywords related to the five variables (e.g. “What was the wettest month in 2023?”) because query interpretation currently uses a keyword-based approach, and timeouts for queries such as “Show temperature patterns decade by decade from 1950 to 2024”, which require lots of triples from the SPARQL endpoint to compile an answer. We are currently exploring LLM-based approaches to increase robustness during interpretation.

3.2. Usage 2: Federated Search

A second value of the EOBS Knowledge Graph is its support for federated search. The agent architecture uses two SPARQL endpoints: the EOBS endpoint as the primary source for climate observations and Wikidata as a secondary source for contextual information. Two federation patterns are supported. In the first pattern, Wikidata is used before querying the EOBS endpoint to resolve place names into geographic coordinates. These coordinates are then matched to the nearest EOBS grid cell. In the second pattern, EOBS query results are enriched after execution with contextual information from Wikidata, such as geographic descriptions or entity metadata. The federation design is fault-tolerant. If Wikidata is unavailable or does not return a result, the EOBS query can still be executed. The system also keeps source attribution explicit: observational values are always retrieved from the EOBS Knowledge Graph and are never replaced by values from Wikidata. This separation is important for maintaining scientific traceability. The example below illustrates the Wikidata → EOBS workflow.

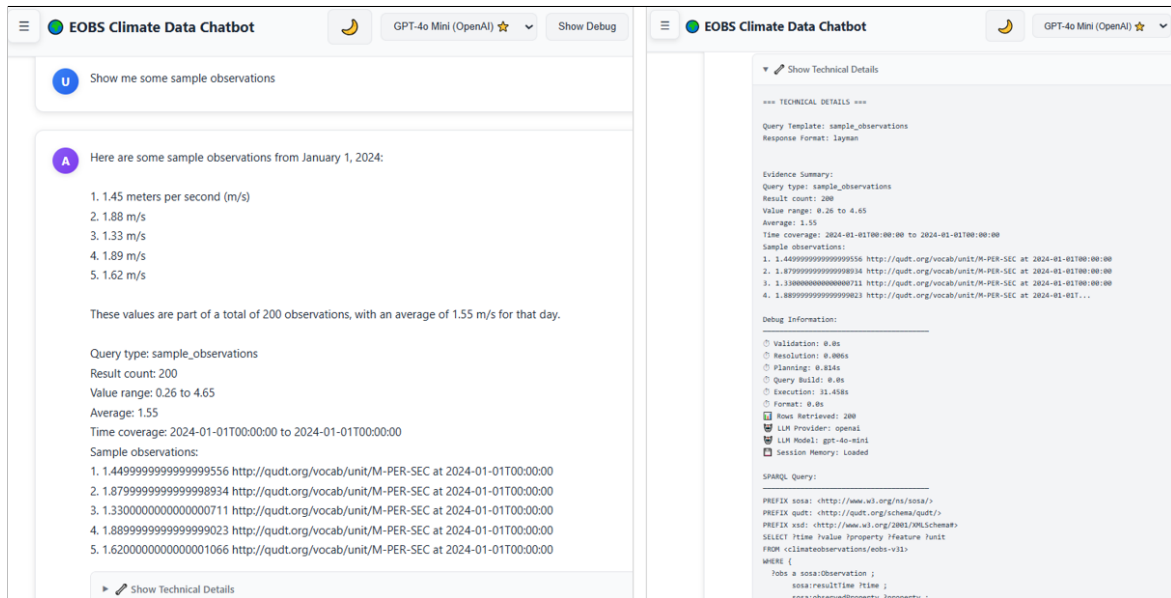


Figure 2: Examples of queries and answers from the LLM Agent. Left: natural language query and answer; Right: expanded menu to see query processing steps and the SPARQL query sent to the triple store.

Table 2

Selected questions by category and SPARQL query templates for the category. The coloring indicates the assignment to the three distinct cognitive operations: **overview**, **zoom and filter**, and **details-on-demand**.

Category	Question/Query	SPARQL Query Template
Overview	What variables are available?	list_properties
Overview	List all locations of observations available	list_features_of_interest
Overview	Give me an overview of climate observations for 2024	all_properties_summary
Aggregation (Multi-year)	Show me wind speed trends for the last 5 years	timeseries_statistics
Aggregation (single-year)	Calculate temperature statistics for 2024	timeseries_statistics
Aggregation (mean-year)	What was the average temperature in 2020?	average_for_property_date_range
Aggregation (mean-month)	Show monthly temperature averages for 2022	monthly_aggregates
Aggregation (mean-day)	What are daily humidity averages for March 2000?	daily_aggregates
Nested-aggregation	What was the mean daily temperature in 2001?	monthly_mean_from_daily
Extreme-values	What were the highest temperature values in 2023?	top_extremes_for_property
Subsampling	Show me some sample observations	sample_observations
Filtering (threshold)	Find temperature readings above 30 degrees in Summer 2024	filtered_timeseries
Filtering (range)	Find precipitation values below 12 mm and above 10 mm in 2001	filtered_timeseries
Location-based	What are grids near lat: 67.8, lon: 20.3	features_near_coordinates
Location-based	What was the weather like in France during 1985?	location_based_summary
Location-based	Compare temperature across different grid points in 2019	timeseries_statistics_by_feature

Example of question: “Show me climate data about Berlin during July 2024.” The agent answers the question in two steps. First, Wikidata is used to resolve a place name into geographic coordinates. Second, the EOBS Knowledge Graph is queried for observations at the corresponding grid-cell level.

Step 1: Resolve the place name using Wikidata. The agent queries Wikidata to resolve the location name into geographic coordinates. With “Berlin” as an exemplary place name, the query returns the Wikidata entity Q64, latitude 52.52, longitude 13.38, and the label “Berlin”.

```

SELECT ?item ?itemLabel ?lat ?lon ?sitelinks
WHERE {
  ?item rdfs:label "Berlin"@en .
  VALUES ?instanceClass { wd:Q515 wd:Q486972 wd:Q1093829 wd:Q15284
    wd:Q3957 wd:Q532 wd:Q131168 wd:Q10742 wd:Q1549591
    wd:Q84 wd:Q16110 wd:Q1337818 wd:Q2615557 }
  ?item wdt:P31/wdt:P279* ?instanceClass .
  ?item wdt:P625 ?coords .
  BIND(geof:latitude(?coords) AS ?lat)
  BIND(geof:longitude(?coords) AS ?lon)
  OPTIONAL { ?item wikibase:sitelinks ?sitelinks . }
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
ORDER BY DESC(?sitelinks) LIMIT 1

```

Step 2: Query the EOBS Knowledge Graph. The agent finds the nearest grid to the resolved location (i.e. grid_47p88_13p62) using the coordinates of the grid and the great circle distance. The query below is then executed using cf:air_temperature as an observedProperty, and a FILTER operation to select values between July 1 and July 31, 2024.

```
SELECT (COUNT(?obs) AS ?count) (AVG(?value) AS ?avgTemp) (MIN(?value) AS ?minTemp)
(MAX(?value) AS ?maxTemp) (SAMPLE(?unit) AS ?unit)
WHERE { ?obs a sosa:Observation ;
  sosa:observedProperty cf:air_temperature ;
  sosa:hasFeatureOfInterest eobs:grid_47p88_13p62 ;
  sosa:hasResult ?result;
  sosa:resultTime ?time.
  ?result qudt:numericValue ?value .
  OPTIONAL { ?result qudt:unit ?unit }
  FILTER ( ?time >= "2024-07-01T00:00:00"^^xsd:dateTime && ?time < "2024-08-01T00:00:00"^^xsd:dateTime )}
```

The query shown above returns 93 temperature observations for July 2024. The average temperature is 19.92 °C, the minimum temperature is 9.64 °C, and the maximum temperature is 30.36 °C. The unit is unit:DEG_C. The agent then verbalizes the result as: *“Found nearest observation station ‘grid_47p88_13p62’ (516.2 km from Berlin) via EOBS geometry graph. During July 2024 in Berlin, the average air temperature was 19.93 °C, with a range from 9.64 °C to 30.36 °C (n=93). The average precipitation amount was 3.27 mm, with a range from 0.00 mm to 13.10 mm (n=31). The average relative humidity was 77.46%, ranging from 67.10% to 92.50% (n=31).”* In sum, this example demonstrates a Wikidata → EOBS workflow: Wikidata supports geographic interpretation of the user input, while the climate measurements are retrieved from the EOBS Knowledge Graph.

4. Conclusion and Future Work

This work has introduced EOBS-KG, a knowledge graph (3.2 billion triples) of European climate observations covering the time period 1950-2024. The usefulness of the graph was highlighted through the EOBS-Agent, an LLM-based Agent that enables natural-language querying of the graph and an example of federated search involving Wikidata. A limitation worth mentioning at this point is that while the pipeline to generate triples from NetCDF data is automated, the first step of downloading the NetCDF data is still a manual endeavor, as the datasets are only available through the e-obs website at the moment. Future work can explore going beyond the simple questions answered currently to build agents to answer more complex analytical questions on European climate observations.

Data and Software availability. A dump of the EOBS Knowledge Graph is accessible at <https://doi.org/10.34740/kaggle/dsv/18498229>. The whole graph can also be queried through a triple store at <https://w3id.org/climateobservations/eobs-v31>. The script for the transformation pipeline (NetCDF to RDF turtle) can be found at <https://github.com/ammara450/eobs-pipeline>. The script for the EOBS Agent can be accessed at <https://github.com/ammara450/climate-chat-agent>, and a short demo of the agent (Version 1: August 2026) can be found at <https://youtu.be/Xo6CJ-zXorw>.

Acknowledgments. We acknowledge the E-OBS dataset from the EU-FP6 project UERRA (<http://www.uerra.eu>) and the Copernicus Climate Change Service, and the data providers in the ECA&D project (<https://www.ecad.eu>). This work has been funded by the European Commission through the project EOSC Data Commons (<https://doi.org/10.3030/101188179>) and the German Research Foundation (DFG) through the project NFDI4Earth (DFG project no. 460036893, <https://www.nfdi4earth.de/>). We thank Ralf Klammer for his assistance with the underlying infrastructure for the triple store.

Declaration on Generative AI. During the preparation of this work, the authors used ChatGPT and QuillBot to assist with grammar and spelling checks. All content was subsequently reviewed and revised by the authors, who take full responsibility for the final publication.

References

- [1] J. Z. Pan, S. Razniewski, J.-C. Kalo, S. Singhanian, J. Chen, S. Dietze, H. Jabeen, J. Omeliyanenko, W. Zhang, M. Lissandrini, R. Biswas, G. de Melo, A. Bonifati, E. Vakaj, M. Dragoni, D. Graux, Large Language Models and Knowledge Graphs: Opportunities and Challenges, *Transactions on Graph Data and Knowledge (TGDK)* 1 (2023). doi:10.4230/TGDK.1.1.2.
- [2] M. R. A. H. Rony, Y. Zuo, L. Kovriguina, R. Teucher, J. Lehmann, Climate Bot: A machine reading comprehension system for climate change question answering, in: L. D. Raedt (Ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI 2022)*, ijcai.org, Vienna, Austria, 2022, pp. 5249–5252. doi:10.24963/IJCAI.2022/729.
- [3] S. A. Vaghefi, D. Stambach, V. Muccione, J. Bingler, J. Ni, M. Kraus, S. Allen, C. Colesanti-Senni, T. Wekhof, T. Schimanski, G. Gostlow, T. Yu, Q. Wang, N. Webersinke, C. Huggel, M. Leippold, ChatClimate: Grounding conversational AI in climate science, *Communications Earth & Environment* 4 (2023) 480. doi:10.1038/s43247-023-01084-x.
- [4] Z. Chen, X. Wang, Y. Liao, M. Lin, M. Zang, Z. Wang, Y. Bai, Enhancing large language models through instruction tuning for answering climate change questions, in: *IGARSS 2025 - 2025 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, Brisbane, Australia, 2025, pp. 2299–2303. doi:10.1109/IGARSS55030.2025.11243408.
- [5] G. Mai, K. Janowicz, R. Zhu, L. Cai, N. Lao, Geographic question answering: challenges, uniqueness, classification, and future directions, in: P. Partsinevelos, P. Kyriakidis, M. Kavouras (Eds.), *Proceedings of the 24th AGILE Conference on Geographic Information Science (AGILE 2021)*, 2021, pp. 1–21. doi:10.5194/agile-giss-2-8-2021.
- [6] K. Janowicz, F. van Harmelen, J. A. Hendler, P. Hitzler, Why the data train needs semantic rails, *AI Magazine* 36 (2015) 5–14. doi:10.1609/aimag.v36i1.2560.
- [7] S. Lumley, R. Sieber, R. Roth, A framework and comparative analysis of web-based climate change visualization tools, *Computers & Graphics* 103 (2022) 19–30. doi:10.1016/j.cag.2021.12.007.
- [8] L. Zepner, P. Karrasch, F. Wiemann, L. Bernard, ClimateCharts.net – an interactive climate analysis web platform, *International Journal of Digital Earth* 14 (2021) 338–356. doi:10.1080/17538947.2020.1829112.
- [9] netcdf4, netCDF4 API Documentation, 2026. URL: <https://unidata.github.io/netcdf4-python/>, accessed: May 2026.
- [10] R. C. Cornes, G. Van Der Schrier, E. J. M. Van Den Besselaar, P. D. Jones, An ensemble version of the E-OBS temperature and precipitation data sets, *Journal of Geophysical Research: Atmospheres* 123 (2018) 9391–9409. doi:10.1029/2017JD028200.
- [11] ecad.eu, E-OBS gridded dataset, 2026. URL: <https://www.ecad.eu/download/ensembles/download.php>, accessed: May 2026.
- [12] R. Rew, G. Davis, NetCDF: an interface for scientific data access, *IEEE Computer Graphics and Applications* 10 (1990) 76–82. doi:10.1109/38.56302.
- [13] M. Ceriani, F. Ciroku, A. Russo, M. Schembri, F. Fung, N. Mittal, V. Trianni, A. G. Nuzzolese, The climate change knowledge graph: supporting climate services, 2026. doi:10.48550/arXiv.2602.19786, arXiv:2602.19786 [cs].
- [14] J. Wu, F. Orlandi, D. O’Sullivan, S. Dev, LinkClimate: An interoperable knowledge graph platform for climate data, *Computers & Geosciences* 169 (2022) 105215. doi:10.1016/j.cageo.2022.105215.
- [15] X. Ma, J. G. Zheng, J. C. Goldstein, S. Zednik, L. Fu, B. Duggan, S. M. Aulenbach, P. West, C. Tilmes, P. Fox, Ontology engineering in provenance enablement for the National Climate Assessment, *Environmental Modelling & Software* 61 (2014) 191–205. doi:10.1016/j.envsoft.2014.08.002.
- [16] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, M. Lefrançois, SOSA: A lightweight ontology for sensors, observations, samples, and actuators, *Journal of Web Semantics* 56 (2019) 1–10. doi:10.1016/j.websem.2018.06.003.
- [17] FAIRsharing Team, FAIRsharing record for: Quantities, Units, Dimensions and Types, 2015. doi:10.25504/FAIRSHARING.D3PQW7.

- [18] OGC, GeoSPARQL Ontology, 2026. URL: <http://www.opengis.net/ont/geosparql>, accessed: May 2026.
- [19] J. R. Herring, OpenGIS[®] Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture, Technical Report, Open Geospatial Consortium, 2011.
- [20] CF Community, CF Standard Name Table, 2026. URL: <https://cfconventions.org/Data/cf-standard-names/current/build/cf-standard-name-table.html>, accessed: May 2026.
- [21] M. B. J. Mark Schildhauer, OBOE: the Extensible Observation Ontology, version 1.2, 2016. URL: https://github.com/NCEAS/oboe/tree/OBOE_1_2. doi:10.5063/F1125R0F.
- [22] L. Kovriguina, R. Teucher, D. Radyush, D. Mouromtsev, SPARQLGEN: one-shot prompt-based approach for SPARQL query generation, in: N. Keshan, S. Neumaier, A. L. Gentile, S. Vahdati (Eds.), Proceedings of the Posters and Demo Track of the 19th International Conference on Semantic Systems (SEMANTICS 2023), Leipzig, Germany, CEUR Workshop Proceedings, CEUR-WS.org, 2023.
- [23] H. M. Zahera, M. Ali, M. A. Sherif, D. Moussallem, A.-C. Ngonga Ngomo, Generating SPARQL from natural language using chain-of-thoughts prompting, in: A. Salatino, M. Alam, F. Ongena, S. Vahdati, A.-L. Gentile, T. Pellegrini, S. Jiang (Eds.), Proceedings of the 20th International Conference on Semantic Systems (Semantics 2024), IOS Press, Amsterdam, The Netherlands, 2024. doi:10.3233/SSW240028.
- [24] J. D'Abramo, A. Zugarini, P. Torroni, Investigating large language models for Text-to-SPARQL generation, in: Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing (KnowledgeNLP'25), Association for Computational Linguistics, Albuquerque, New Mexico, USA, 2025, pp. 66–80. doi:10.18653/v1/2025.knowledgenlp-1.5.
- [25] A. Degbelo, A snapshot of ontology evaluation criteria and strategies, in: R. Hoestra, C. Faron-Zucker, T. Pellegrini, V. de Boer (Eds.), Proceedings of the 13th International Conference on Semantic Systems (SEMANTICS 2017), ACM Press, Amsterdam, The Netherlands, 2017, pp. 1–8. doi:10.1145/3132218.3132219.
- [26] M. Grüninger, M. S. Fox, Methodology for the design and evaluation of ontologies, in: Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Quebec, Canada, 1995.
- [27] B. Shneiderman, The eyes have it: a task by data type taxonomy for information visualizations, in: Proceedings of the IEEE Symposium on Visual Languages, IEEE Computer Society Press, Boulder, Colorado, USA, 1996, pp. 336–343. doi:10.1109/VL.1996.545307.
- [28] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, I. Stoica, Judging LLM-as-a-judge with MT-bench and chatbot arena, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023 (NeurIPS 2023), New Orleans, Louisiana, USA, 2023.