Wikidata Atlas: Putting Wikidata on the Map

Benjamín del Pino benjamin.delpino@ing.uchile.cl DCC, Universidad de Chile Santiago, Chile

ABSTRACT

Wikidata Atlas is an online system that allows users to explore Wikidata items on an interactive global map; for example, users can explore the global distribution of all lighthouses described by Wikidata. Designing such a system poses challenges in terms of scalability, where some classes have hundreds of thousands of instances; efficiency, where visualisations are generated live; freshness, where we want changes on Wikidata to be reflected as they happen in the system; and usability, where we aim for the system to be accessible for a broad audience. Herein we describe the design and implementation of the system in light of these challenges.

CCS CONCEPTS

• Information systems \rightarrow Browsers; Geographic information systems.

KEYWORDS

geographic data, Wikidata, user interfaces

ACM Reference Format:

Benjamín del Pino and Aidan Hogan. 2023. Wikidata Atlas: Putting Wikidata on the Map. In *Companion Proceedings of the ACM Web Conference 2023* (WWW '23 Companion), April 30-May 4, 2023, Austin, TX, USA. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3543873.3587356

1 INTRODUCTION

Wikidata [7] is an open knowledge graph providing structured data relating to over 100 million items. Users can access data about individual items via autocompletion-based search, while more expert uses can query data spanning the knowledge graph via the Wikidata Query Service [4]. While the latter service supports SPARQL, an expressive query language, its user must be knowledgeable about this language. A key challenge for Wikidata is then to engage a broader audience of users by providing them novel and accessible ways to browse and interact with the knowledge graph as a whole.

The system proposed in this paper – called *Wikidata Atlas* – aims to provide both expert and non-expert users with a novel geographical perspective on Wikidata. The system supports visualising Wikidata's geo-referenceable entities (or *geo entities*, for short; i.e., entities with geo-coordinates) on a world map at a global scale. Such items include not only the tangible entities that might immediately spring to mind, such as *cities*, *nuclear power plants*,

WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA

© 2023 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9419-2/23/04.

https://doi.org/10.1145/3543873.3587356

Aidan Hogan ahogan@dcc.uchile.cl IMFD; DCC, Universidad de Chile Santiago, Chile

Table 1: Map-based systems for multi-domain KBs

System	Source	Search	Scope	
DBpedia Mobile [1]	DBpedia	Proximity, Category	Local	
OpenStreetBrowser ^a	OSM, Wikipedia	Category	City	
Wiki Atlas ^b	Wikipedia	Category, Instance	Country	
Wikidata QS ^c	Wikidata	Query	World	
WorldKG [3] ^d	OSM, Wikidata, DBpedia	Query	World	
Wikidata Atlas ^e	Wikidata	Class	Global	

a https://www.openstreetbrowser.org/

^b https://wiki-atlas.org/

c https://query.wikidata.org/

d https://www.worldkg.org/sparql e https://wdatlas.dcc.uchile.cl/

shipwrecks, etc., but also intangible entities (often events) such as *assassinations*, *jazz festivals*, *naval battles*, *nuclear weapons tests*, etc.

Specifically, Wikidata Atlas enables users to search for a class (currently in English) via autocompletion, whose instances are visualised on a map. Given that there can be thousands (or more) of instances, clusters of geographically-near items are presented. As the user zooms in, these clusters divide into individual points. Upon clicking an individual point, a user is presented with a more detailed description of the entity, including its label, Wikidata identifier (with link), coordinates, and, if available, an image and description.

We foresee a number of use-cases for Wikidata Atlas, generally centred around the notion of *exploratory search* [5] through a geographic perspective: the user may not start off with a specific information need, but through exploration, may serendipitously attain new knowledge; for example, a user browsing *nuclear weapons tests* might be surprised to find 1,185 instances in the Nevada desert. Wikidata Atlas can also be used to identify data quality issues, for example, to identify regions of the world with sparse items due to geographical bias; to identify potentially incorrect coordinates, e.g., inland shipwrecks, off-shore cities, or beaches not on a coast; etc.

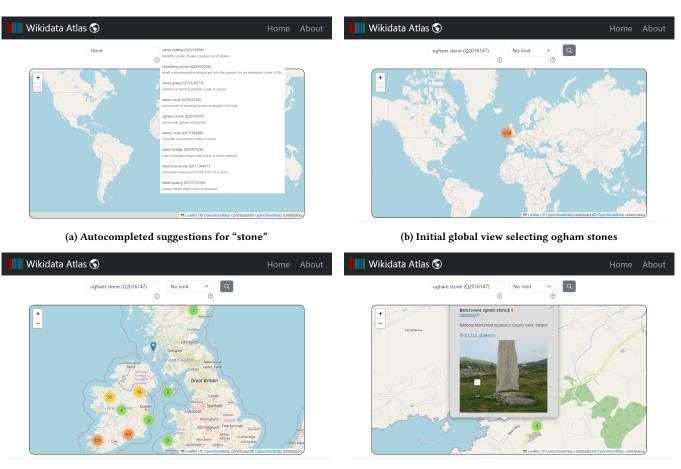
In the design of Wikidata Atlas, we face a number of challenges. SCALABILITY: Millions of Wikidata entities have geo-coordinates, while specific classes like *mountains* have hundreds of thousands of geo instances. EFFICIENCY: We aim for an interactive map generated online; where it is important for the system to be efficient and responsive. FRESHNESS: While scalability and efficiency can be improved by locally caching data, this risks returning stale results as Wikidata changes. USABILITY: Our goal is to offer both expert and non-expert users a novel perspective on Wikidata.

Before describing the design and implementation of Wikidata Atlas in light of these challenges, we first discuss related works.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

del Pino and Hogan

WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA



(c) Zooming in, the cluster divides (d) Detailed view on one instance Figure 1: Interactions exploring ogham stones

2 RELATED WORKS

Table 1 lists related online tools with map-based visualisations of open, multi-domain knowledge-bases (DBpedia, Wikipedia, Wikidata, etc.). DBpedia mobile [1] is an app that can visualise DBpedia entities close to the user's location. OpenStreetBrowser provides an interactive map of entities from OpenStreetMap (OSM) enriched with Wikipedia information up to a city-level scope. Wiki Atlas is an interactive map that visualises all Wikipedia entities up to a sub-continental scope. The Wikidata Query Service (QS) provides a world-map for query results with geo-coordinates. WorldKG [3] combines OSM, Wikidata and DBpedia and hosts a SPARQL query service that displays an interactive map for up to 100 results.

In terms of novelty, unlike DBpedia Mobile, OpenStreetBrowser and Wiki Atlas, which focus on localities, Wikidata Atlas permits visualising entities at a global/world view. While the query services of Wikidata and WorldKG provide a global view, they require the user to know SPARQL; Wikidata Atlas rather offers a more straightforward and accessible but less expressive user interaction.

3 WIKIDATA ATLAS

We now describe the design and implementation of the Wikidata Atlas system, starting with an explanation of the user interaction, moving on to the identification of relevant *geo classes*, the design of the web application, and finally a discussion of scalability.

User Interaction. Figure 1 presents an example user interaction¹ inspired by the use-case described by Thiery et al. [6] relating to *ogham stones*, where an early form of writing – called ogham – was carved into stones spread over Ireland and parts of Britain. In Figure 1a, the user enters "stone" in an autocomplete field and receives a list of suggested classes. If the user chooses the fifth option, ogham stone, they will receive the view shown in Figure 1b, where we see 1,259 instances of ogham stones clustered around the Irish Sea. Zooming in further, per Figure 1c, the cluster will divide into smaller clusters and (isolated) individual points. Clicking an individual point will then bring up a detailed view, with the entity label and Wikidata identifier, and if available, the flag of the country it pertains to, its description, and its image.

Geo-referenceable classes. Per Figure 1a, when the user types "stone", they are suggested what we call *geo-referenceable classes* from Wikidata, or *geo classes* for short, which we define as any value of the instance of (P31) property on Wikidata whose subject has some value for the coordinate location (P265) property. This

¹Screenshots have been graphically modified to improve readability.

Wikidata Atlas: Putting Wikidata on the Map

WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA



Figure 2: Visualisation of 122,852 railway stations

requires custom autocompletion: for the same string, Wikidata's own autocompletion suggests entities that are neither classes nor geo-referenceable, such as Stone a family name, an English town, a 2010 film, etc. Trying to extract all geo classes via the Wikidata QS times out as it requires a join between P31 and P265, both of which have many triples. We thus extract geo classes from a Wikidata dump using local scripts to perform a hash join. Some geo classes suggested by this method tend to be irrelevant, such as family name (Q101352), which has 23 instances with coordinates and 533,205 without: requiring the class to have at least one geo instance is too liberal. Thus, for each geo class in the dump, we further count and store how many instances have coordinates, and how many do not, filtering classes with fewer than 1% of geo instances. We also filter classes without labels in English. For the remaining geo classes, we extract their label (and if available, their description), which will be cached locally along with the aforementioned statistics for autocompletion. The classes suggested by autocompletion are further ranked by the number of geo instances it has.

Web application. The back-end of the web application (built with Flask) handles the autocompletion of geo classes, and the communication with Wikidata. When a geo class such as ogham stone is selected, the back-end sends a query to the Wikidata QS to request all instances of the class and its transitive sub-classes, their coordinates, and if available, their English label and description, their image, and their country code (used to display the flag). The query adds constraints to ensure that the coordinates are on Earth (filtering, e.g., the Olympus Mons on Mars). The query uses string functions to transform the image URL in order to request a lowerresolution image from Wikimedia to avoid wasting bandwidth and speed up the loading of images. The back-end then provides a HTTP API in order to communicate with a client-side front-end written in Javascript using React Leaflet to generate the map visualisation.

Scalability. Some geo classes have hundreds of thousands of instances; while such classes are few in number, these tend to be well-known and thus likely targets for queries (*mountain, street, village, hotel,* etc.). Though the system can scale to hundreds of thousands of instances – see Figure 2 for a map of over one hundred thousand railway stations – extended delays, QS timeouts and unresponsiveness begin to occur at this order of magnitude. It is

challenging to address this issue in a general way. While instance data for such classes could be extracted from the dump and cached to avoid the QS timeouts, such instance data are more dynamic than the list of geo classes cached previously (see Section 4). Also, the map visualisation is run in the user's browser, where problems may arise at different scales depending on the client's setup. Our solution is to give clients the *option* to limit the number of results returned. Furthermore, upon selecting a geo class with more than 10,000 instances, a warning recommends the application of a limit. The limit is passed as a LIMIT clause on the SPARQL query, which further helps to avoid timeouts; however, the results returned are not random, but rather tend to be in order of *n* for the identifier *Qn*, and may thus be biased. Though SPARQL supports ordering, sampling, etc., this internally requires enumerating all results first, increasing timeouts, and thus we opt not to use such features.

4 EVALUATION

We now present our evaluation, which aims to address the challenges mentioned at the outset, namely SCALABILITY, EFFICIENCY, FRESHNESS, and USABILITY. We use the gzipped version of the Wikidata truthy dump from October 22nd, 2022.

Scalability & Efficiency. Extraction of the geo classes from the dump took 3.3 hours on an Acer Aspire ES1-572 notebook, with Intel[®] CoreTM i3-6100U@2.30GHz, 8GB of RAM, SSD hard-disk running Windows 10. We found 24,099 geo classes, of which 23,351 had at least 1% of geo instances. Filtering geo classes with no English label reduced the set to 21,263. Table 2 lists the ten classes with the most such geo instances, along with the number of its instances that were not geo-referenceable, and the percentage that were. We further tested the efficiency of the autocompletion over these classes with all strings formed from [a-z] of length one, two and three; for length one, the median time was 50 ms, while the max was 239 ms; for length two, the median time was 22 ms, while the max was 126 ms; for length three, the median time was 20 ms, while the max was 704 ms; we deem these times to be acceptable. Finally, we investigated the most costly procedure: for each of the 21,263 geo classes, we ran the corresponding query on the Wikidata QS to extracts its (transitive) instances and their metadata. Of these queries, 139 (0.65%) timed out and 6 (0.03%) threw a server error. In Figure 3, we provide a scatter plot for the remaining times, where x indicates the number of results, and y the runtime; to reduce the number of points, we average the runtime for each unique value of *x*. From this plot we can see a loose positive correlation between the number of results and the runtimes. Most points occur in the bottom left of Figure 3; specifically, the min, median, mean, and max runtimes were 0.56 s, 1.26 s, 2.59 s, and 138.08 s, respectively.

Freshness. The local cache of geo classes and their statistics may become stale over time and refreshing them from the dump takes several hours (including the download). To analyse how this affects freshness, we took a dump from May 22th, 2022 and compared the data cached vs. October 22th, 2022 (a gap of 5 months). We found an increase of 4% new geo classes. However, these were relatively obscure classes. Since instance data are always fetched live from Wikidata, we think that irregular (e.g., monthly) updates of the cached data are sufficient to avoid staleness for typical usage.

WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA

Table 2: Classes with most geo-referenceable instances

Label	Geo	⊐Geo	%Geo
mountain	520,636	6,976	98.7
human settlement	434,470	137,774	75.9
street	413,646	212,672	66.0
river	367,245	30,805	92.3
hill	321,447	6,483	98.0
lake	265,975	6,447	97.6
building	252,311	15,124	94.3
house	201,672	11,800	94.5
church building	200,881	26,818	88.2
village	180,134	55,353	76.5
	human settlement street river hill lake building house church building	human settlement 434,470 street 413,646 river 367,245 hill 321,447 lake 265,975 building 252,311 house 201,672 church building 200,881	human settlement 434,470 137,774 street 413,646 212,672 river 367,245 30,805 hill 321,447 6,483 lake 265,975 6,447 building 252,311 15,124 house 201,672 11,800 church building 200,881 26,818

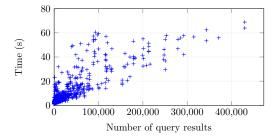


Figure 3: Mean runtimes vs. results returned per class

Usability. In order to gain initial insights into the usability of the system, we created an online questionnaire that asked user to perform four queries - stadium, metro station, clandestine center of detention and torture, educational organization - hinting in the last case that they may wish to apply a limit. They then responded to a System Usability Scale (SUS) survey [2], rating ten claims about the system on a Likert scale from 1 (disagree) to 5 (agree). The SUS score ranges from 0-100, where a score of 68 is considered average. A Spanish version of the questionnaire was shared on a forum of students and professors at the Department of Computer Science, University of Chile, receiving 16 responses. An English version was shared on the public Wikidata mailing list, receiving 2 responses. Table 3 presents the mean and standard deviation of responses for each claim, along with the SUS score. Considering all 18 respondents, the overall SUS score was 82.92 ± 10.16 . A comment field was left at the end, where users noted that the system was quite simple and intuitive. One user noted surprise about some suggested classes (like singer) that did seem geographical, and another user noted that classes relating to hate crimes had few results. One user noted that search was unintuitive (mentioning the limit feature), and another user noted their opinion that searching globally was of limited use compared with searching by locality. Users requested multilingual support, shareable URLs with query parameters, and displaying more information in the detail view.

5 CONCLUSIONS

We have presented Wikidata Atlas: a system for exploring instances of a specified Wikidata class via an interactive world map. We believe that this system is useful for exploratory search, and hope it might give both expert and non-expert users a new perspective on the Wikidata knowledge graph. In terms of future work, a key point would be to address the user feedback we collected, and in particular, to offer multilingual support, and browsing of all entities in a given del Pino and Hogan

Table 3: SUS results (m: mean; s: standard deviation)

Claim to evaluate	ES (m	16) s	EN m	(2) s	All m	(18) s
I think that I would like to use this system frequently.		1.10	2.50	2.12	3.83	1.25
I found the system unnecessarily complex		1.00	2.00	0.00	1.94	0.94
I thought the system was easy to use.		0.52	4.50	0.71	4.50	0.51
I think that I would need the support of a technical person to be able to use this system.		0.58	1.00	0.00	1.22	0.55
I found the various functions in this system were well integrated.		1.18	2.50	0.71	3.78	1.22
I thought there was too much inconsistency in this system.		0.73	2.50	0.71	1.56	0.78
I would imagine that most people would learn to use this system very quickly.		0.96	4.00	0.00	4.39	0.92
I found the system very cumbersome to use.		0.63	2.00	1.41	1.61	0.70
I felt very confident using the system.		0.73	3.50	0.71	4.39	0.78
I needed to learn a lot of things before I could get going with this system.		0.62	1.50	0.71	1.39	0.61
SUS Score	84.53	9.41	70.00	7.07	82.92	10.16

locality. We are also eager to explore other visual paradigms for plotting instances, such as a heat-map of countries per their number of instances. It would also be interesting to investigate new querying methods that find a trade-off between the simplicity of classes and the expressivity of a SPARQL query; perhaps a faceted selection of instances would be interesting, for example, to display *lighthouses made from brick on the North sea*. Finally, a more extensive usability study would be useful to generate additional feedback and insights.

Supplemental material. A demo of the system is available at https://wdatlas.dcc.uchile.cl/. Source code is available at https://github.com/benjamindpb/wdatlas.

ACKNOWLEDGMENTS

This work was supported by Fondecyt Project No. 1221926 and ANID – Millennium Science Initiative Program – Code ICN17_002.

REFERENCES

- Christian Becker and Christian Bizer. 2009. Exploring the Geospatial Semantic Web with DBpedia Mobile. J. Web Semant. 7, 4 (2009), 278–286. https://doi.org/10. 1016/j.websem.2009.09.004
- [2] John Brooke. 1996. SUS A quick and dirty usability scale. Usability Evaluation in Industry 189, 194 (1996), 4–7.
- [3] Alishiba Dsouza, Nicolas Tempelmeier, Ran Yu, Simon Gottschalk, and Elena Demidova. 2021. WorldKG: A World-Scale Geographic Knowledge Graph. In CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021. ACM, 4475-4484. https://doi.org/10.1145/3459637.3482023
- [4] Stanislav Malyshev, Markus Krötzsch, Larry González, Julius Gonsior, and Adrian Bielefeldt. 2018. Getting the Most Out of Wikidata: Semantic Technology Usage in Wikipedia's Knowledge Graph. In The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, Oct. 8-12, 2018, Proc., Part II (LNCS, Vol. 11137). Springer, 376–394.
- [5] Émilie Palagi, Fabien Gandon, Alain Giboin, and Raphaël Troncy. 2017. A Survey of Definitions and Models of Exploratory Search. In Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics, ESIDA@IUI 2017, Limassol, Cyprus, March 13, 2017. ACM, 3–8. https://doi.org/10.1145/3038462. 3038465
- [6] Florian Thiery, Sophie C. Schmidt, Timo Homburg, and Martina Trognitz. 2020. The SPARQL Unicorn: An introduction. Zenodo.
- [7] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. Commun. ACM 57, 10 (2014), 78–85.

Received 3 February 2023