DataQuest: Web Augmentation with Wikidata

Diego Pizarro[1], Sergio Firmenich[2] and Aidan Hogan[1]

[1]DCC, Universidad de Chile & IMFD
[2]Universidad Loyola Andalucía - Spain

Abstract

In this demo paper we present DataQuest: a browser extension that leverages the Wikidata knowledge graph to augment Web navigation. While visiting a webpage in the browser that matches an external identifier registered in Wikidata, the DataQuest extension queries for information about the associated entity in Wikidata and uses this to display additional information about the entity and guide the user's next navigation steps. We describe the design and implementation of the DataQuest extension, and present a preliminary user evaluation to gain some initial insights regarding its performance and usability.

1. Introduction

The Web has expanded and evolved considerably since its inception in 1989. However, the way in which users navigate the Web has not changed all that much since the first browsers were released in the early 90s. Users enter the URL of a webpage, browse its content, and either follow links or fill forms provided in the webpage to navigate to related content of interest. Researchers have proposed various techniques down through the years with which to augment the Web, and how users interact with it [1], but these have had limited impact on how billions of users navigate the modern Web, which remains largely unchanged.

In parallel with these efforts to augment Web navigation, initiatives relating to the Semantic Web have sought to enable software agents to navigate the Web on the user's behalf [2]. A relatively recent development in this direction has been the publication of open knowledge graphs, such as Wikidata [3]. Wikidata is an open knowledge graph collaboratively-edited by thousands of users. It currently describes more than 100 million entities from various domains.

In the context of Web augmentation, an interesting feature of Wikidata is its provision of a diverse set of external identifiers for entities via properties such as IMDb ID (P345), which links movie-related entities in Wikidata to their corresponding pages on the Internet Movie Database (IMDb) website. As of the time of writing in August 2025, Wikidata features 3,575 external identifier properties providing 135,431,998 links to external webpages.[1] While these are often used to navigate from Wikidata to external sources of data about an entity, they can also be used to retrieve the Wikidata entity and related information for the entity that an external (linked) website describes. Put more simply, such links provide an entity-centric and unambiguous bridge between the broader Web and the Wikidata knowledge graph.

Wikidata'25: Wikidata Workshop at ISWC 2025
diego.pizarro0@gmail.com (D. Pizarro); sdfirmenich@uloyola.es (S. Firmenich); ahogan@dcc.uchile.cl (A. Hogan)
0000-0001-9502-2189 (S. Firmenich); 0000-0001-9482-1982 (A. Hogan)

[1]See https://www.wikidata.org/wiki/Wikidata:Database_reports/External_identifier_completeness

We argue that open knowledge graphs can be leveraged to augment the Web, and to help users to better navigate the Web towards achieving their goals. A natural starting point is to consider how Wikidata and suitable tooling may help a user while browsing the Web, wherein we propose the following forms of assistance:

- By offering complementary metadata extracted from Wikidata about the entity described by the current webpage.
- By offering navigation steps via "virtual links" derived from Wikidata; for example:
  - By navigating from a webpage, through Wikidata's knowledge graph, onto a related entity, and then onto a webpage about that related entity on the same site.
  - By quickly navigating between webpages on different websites describing the same entity (via Wikidata's external identifiers for that entity).
  - By offering navigation steps induced from Wikidata, for example, to navigate to webpages of similar entities – with similarity calculated from Wikidata – on the current website or a different website.

In this demo paper, we introduce DataQuest: a browser extension that implements these ideas, and thus provides an initial, concrete demonstration of the potential for Wikidata to enrich the user's navigation of the Web in novel ways.

Motivating scenario Alice is looking for a movie to watch tonight on a streaming platform and is in the mood for a dystopian sci-fi like *Blade Runner*. She visits the *Blade Runner* IMDb page and then visits the director's page (Ridley Scott) to see what other movies he directed along similar lines, where *The Martian* catches her eye. Visiting its IMDb page, she is interested to know if it won any notable awards, but the information is not presented on the page. She uses DataQuest to pull data about the movie from Wikidata, where she sees it has won a Hugo Award, and has Academy Award nominations. The IMDb rating is good, so she uses DataQuest to navigate directly to other popular sites to check the ratings of *The Martian* there, such as Metacritic, Rotten Tomatoes, etc., which are also promising (the links are taken from Wikidata via the entity for *The Martian* (Q18547944)). However, reading a review, she realizes that she has already seen the movie. Hence she uses DataQuest to find similar movies (calculated from Wikidata), which suggests the movie *Gravity*. Interested in streaming this movie, she uses DataQuest to jump to the Netflix page (pulled from Wikidata via the Netflix ID (P1874)).

Paper structure The rest of the paper is structured as follows:

- Section 2 presents related works on Web augmentation.
- Section 3 presents the design and implementation of the DataQuest browser extension.
- Section 4 presents a preliminary evaluation of DataQuest's performance and usability.
- Section 5 concludes the paper.

2. Related Work

We now present related works that are increasingly specific and closer in topic to our contribution. First, we provide a broad overview of Web augmentation; then we discuss proposals for such augmentation that leverage the Semantic Web, and, finally, Wikidata.

Web augmentation Web augmentation [1] is a set of techniques and tools that aim to improve the user's experience while using web applications without relying on the owner of the application. Although there are several ways to achieve this, probably the most convenient is by means of web browser extensions. These software artifacts may be aware of the applications in use by the user and may manipulate them to add, remove, or modify content and functionality. Several approaches have emerged over the last two decades with different purposes and intentions [4], such as improving accessibility, content integration (in a mashup style), supporting frequent tasks, personalization, etc. Other approaches around this idea are focused specifically on navigation, such as the case of a client-side framework for concern-sensitive navigation [5]. We refer the reader to a recent survey on Web augmentation for more details [4].

Web augmentation via the Semantic Web One might argue that the goals of the Semantic Web and Web augmentation are analogous, but in practice the types of approaches investigated by both communities are distinct: the Semantic Web focuses on structuring the content of the Web to make it more machine readable, while Web augmentation points towards techniques and tooling to improve the user experience on the human-readable Web. We find few works in this intersection but one such work proposes SWAX [6], which allows end-users without advanced programming skills to build Web augmentation artifacts that takes some information from the current Web page, and produce new related information gathered from the Semantic Web that is at the end woven into the current website. Specifically, upon visiting a webpage from a particular site, the SWAX tool – given a parameterized SPARQL query, a SPARQL endpoint, a path in the DOM of the website, and an output HTML template (generated *a priori*) – fills the elements extracted from the webpage using the DOM path as parameters into the SPARQL query, executes the resulting query on the endpoint, and injects the results directly into the webpage using the HTML template. Other works on semantic annotation of Web pages [7], on semantic content management systems [8, 9], among others, also relate indirectly to Web augmentation. No such work that we are aware of is oriented to assist navigation.

Wikidata-based browser extensions There exist also some web browser extensions related to Semantic Web and more specifically with Wikidata. Wwwyzzerdd [10], for instance, is a project based on web browser extensions that uses information available on Wikipedia and transcribes it into Wikidata when such information does not already exist there. The extension uses the Wikidata API to query the current page and allows modifications to Wikidata based on the information found on that page. Although this extension is not directly related to Web augmentation – but rather Wikidata enhancement – it is worth mentioning as it demonstrates the feasibility of creating browser extensions that interact with Wikidata. Another tool based

also on Wikidata and web browser extensions is Wikidata for Web[2], which is the closest approach to ours, being similar to DataQuest in various aspects. It too uses external identifiers to identify the Wikidata entity relating to the current webpage, it too displays information from Wikidata in a panel, and it too offers quick-links to navigate to pages on other websites that describe the same entity. However, DataQuest is focused on enhancing web navigation, and thus provides navigation features that Wikidata for Web does not cover, such as using Wikidata to more quickly navigate to pages about related entities on a given website, or to navigate to similar entities on the same website. Additionally, DataQuest offers a much more lightweight design than Wikidata for Web, providing a small icon in the extensions tray that only generates external requests when interacted with (further improving user privacy). On the other hand, Wikidata for Web and Wwwyzzerdd provide functionalities that DataQuest does not offer, such as the ability to extract and add information to Wikidata.

Novelty We believe that the integration of well-structured information – such as that obtained from Wikidata – may assist users to better *navigate* websites with unstructured information. Like SWAX and Wikidata for Web, we support displaying information from Wikidata while browsing the web, but unlike these and related tools, our focus is on assisting users to *navigate* the Web (between websites, between pages of the same website, etc.) in a lightweight manner.

3. System design and implementation

We now describe the criteria, design, and implementation of DataQuest.

Criteria In the creation of DataQuest, our focus was on leveraging Wikidata to help end-users navigate the Web in novel ways. We identified the following key criteria:

- *Usability*: The tool is aimed at an end-user audience who is not necessarily familiar with Wikidata or Semantic Web technologies.
- *Responsiveness*: The tool aims to be responsive, avoiding long response times.
- *Privacy*: Requesting information from Wikidata leaves traces in the corresponding API and query service logs, so the user's privacy requires careful consideration.
- *Low footprint*: The tool should not overwhelm Wikidata with many/costly requests.
- *Customizability*: The tool should be configurable to suit a particular user's needs.

There are also some criteria imposed by technical limitations, for example, to minimize the types of permissions required for the extension (also related to privacy).

Disambiguation DataQuest links the current webpage that the user is visiting in their browser to a Wikidata entity via the external identifiers provided by Wikidata. For example, if the user is browsing IMDb, than the webpage will be matched to a Wikidata entity via the IMDb ID (P345) external identifier property. This ensures unambiguous identification of the Wikidata entity described by the current webpage, but with the limitation that no functionality is currently provided if a corresponding external identifier does not exist.

[2] See https://www.wikidata.org/wiki/Wikidata:Tools/Wikidata_for_Web

Design Figure 1 illustrates the initial view of DataQuest, as we now describe.

The user is presented with a small purple circle in the extension icon tray. While browsing the Web, the user can click this circle, which triggers the extension to see if there is a matching Wikidata entity. If there is no matching entity, the icon stays purple and DataQuest currently offers no functionalities. If there is a matching entity, the circle turns green.[3] DataQuest then offers three functionalities as three tabs.

The Info functionality, as illustrated in Figure 1, provides an overview of information from Wikidata, showing available properties in collapsible lists. The user can expand a property to see a list of values (for *Usability* reasons, we currently show only truthy values without references, qualifiers, etc.; these can be retrieved, if needed, by visiting the entity on Wikidata). Since our focus is on navigation, we add links on values for which Wikidata has external links on the *current* website; clicked links open in a new tab. We add a hyperlink icon on properties that have some value with such a navigation option. This tab can thus both inform the user, and provide them navigation options to related entities on the same site.

The Navigation functionality, as illustrated in Figure 2, allows the users to navigate to other websites describing this entity. Though cropped for space reasons, this view is shown in the same context as Figure 1 (over the IMDb page for Ridley Scott); also a scroll on the right-hand side allows the user to choose from dozens of external links further down. These links allow users to navigate to the page describing the same entity on other websites. Hard-coded at the top of the tab are links to Wikimedia sites. Since entities might display many such links, for *Usability*, we provide website icons from Wikidata that allow users to quickly distinguish different links; for *Customizability*, we allow users to "star" quick links that they frequently access, where the Rotten Tomatoes ID is presented first since it was previously starred.

The Similar functionality, as illustrated in Figure 2, and again cropped for space, is intended as a proof-of-concept feature whereby navigation options are computed over Wikidata and offered to the user. Specifically, this tab offers the user a list of the 10 most similar entities to the current focus entity, with similarity computed over Wikidata. We delegate the computation of similarity relations to an external service (described later), where we loosely expect such a relation to correlate with user interest, i.e., the more that entity A is similar to entity B, the more likely that a user interested in entity A is interested in entity B. Upon hovering over a link, a brief preview of that entity is provided (with a label, description and image, as available, via Wikidata). The links navigate to the page for that entity on the same website.

Implementation DataQuest is a Chromium extension built with the Plasmo framework.

To identify the entity associated with the current entity, the extension pulls a list of external property IDs from the Wikidata Query Service (WDQS) [11], along with their formatter URLs (e.g., `https://www.imdb.com/title/$1/`, where `$1` is a placeholder for an IMDb ID, such as `tt0083658`). These are cached for future use, and the URL of the current webpage is matched against this list. To improve *Responsiveness*, we optimize this step using a dictionary where

[3] In the original design of the DataQuest extension, the icon would turn from purple to green automatically as the user browses the Web, indicating that an entity was found on Wikidata. However, this would involve sending a request to Wikidata for every page visited, breaking the criteria of *Privacy* and *Low footprint*. In the current design, the extension only interacts with Wikidata when the extension icon is actively clicked on by the user.

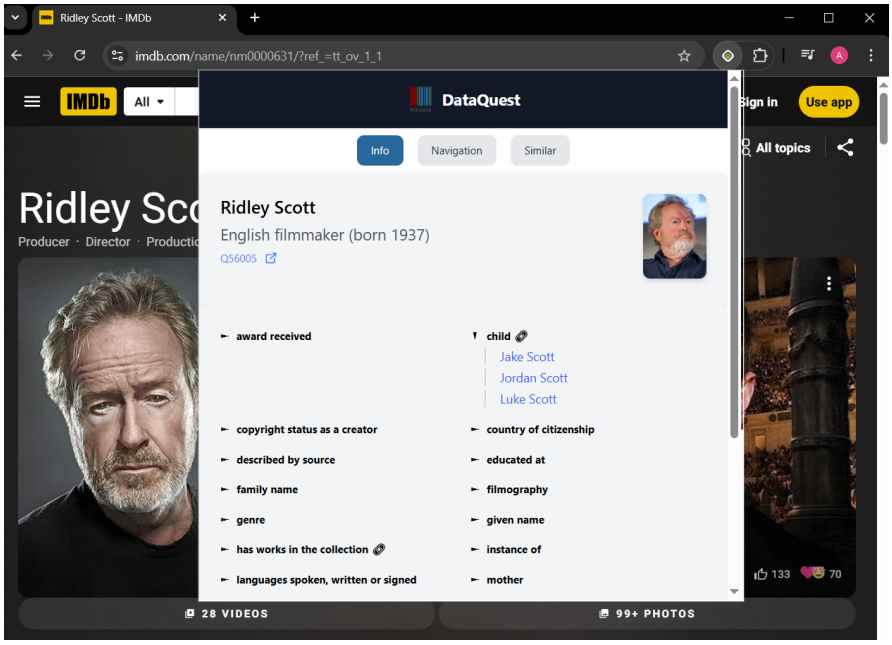

Figure 1: Initial view of DataQuest with the Info tab

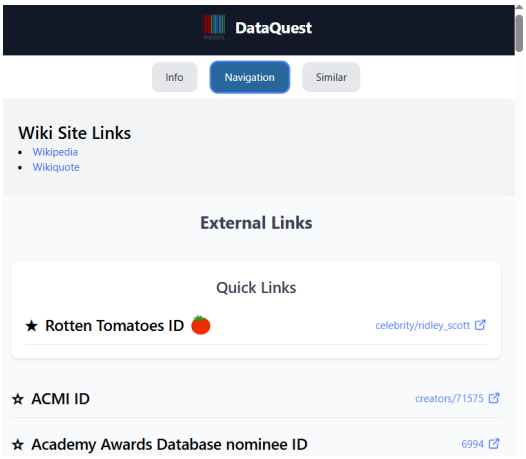

Figure 2: Navigation tab of DataQuest

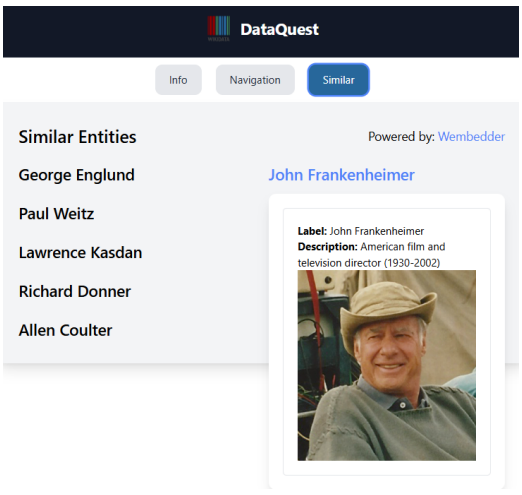

Figure 3: Similar tab of DataQuest

the keys are the domain (e.g., `imdb.com`), and the values are the formatter URLs and their external-ID property. The domain of the current URL is extracted, looked-up in the dictionary, and then only the formatter URLs for that specific domain are checked to see which matches the current URL, extracting the string matching the parameter. If a match is found, this extracted value, along with the external ID property, are used to query WDQS for the entity. This index contains approximately six thousand keys currently.

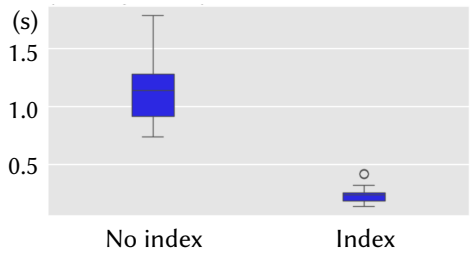

Figure 4: Times for finding Wikidata entity

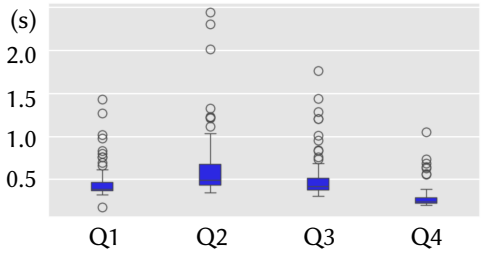

Figure 5: Times for queries over WDQS

Once the Wikidata entity is identified, the information needed to power the Info and Navigation features are queried via WDQS.[4] The Similar functionality is powered by an external similarity service over Wikidata, namely Wembedder [12], which uses knowledge graph embeddings over Wikidata entities to provide an API of k-nn similarity over Wikidata. (While Wembedder serves as a prototype for such functionality, the service is based on a static dump of Wikidata that is several years old, and thus does not cover newer entities.)

The extension has been published in the chrome web store, and is available at the following link: https://chromewebstore.google.com/detail/dataquest/ehalaaaolkejaknfgndjjgdilojeaemd. Source code is available at https://github.com/dpizarrow/dataquest.

4. Evaluation

In the following section, we present a brief, preliminary evaluation of DataQuest from two perspectives: how long queries take to resolve, and the usability of the application.

Performance evaluation We use the examples for formatter URLs given on Wikidata to evaluate how long the different DataQuest steps take, using in total around ten thousand such URLs for the experiment. Figure 4 presents the times for the first step of identifying the Wikidata entity associated with a particular URL (if any), including the query. This step takes 1.12 ± 0.23 seconds on average ($\pm$ indicates standard deviation) without the domain-index optimization (comparing each URL against every formatter URL available on Wikidata) and 0.23 ± 0.05 seconds with the domain-index optimization (comparing the URL only against formatter URLs from the same domain). Figure 5 presents the times for queries over WDQS; namely, Q1: getting the label, description and image of an entity; Q2: for getting the properties and values of an entity; Q3: for getting the Wikimedia sitelinks of an entity; and Q4: for getting links to all websites that describe the entity. Overall, the times are below half a second in the median case, with some outliers up to 2.5 seconds. Querying for similar entities takes 0.3 seconds in the median case, and up to 0.4 seconds. From this analysis, we see that DataQuest is *Responsive*, and its queries have a *Low footprint* for WDQS and Wembedder.

[4]We currently use WDQS rather than the REST API as it offers more customization of the information returned, e.g., only return truthy relations, and (as we discuss later) the performance is sufficient for responsive behavior.

User evaluation To conduct a preliminary user evaluation, we designed a scenario whereby the user must install the extension, navigate to the IMDb *Terminator* page, revise the Info panel and navigate to the IMDb page of the director via the panel, navigate to the Rotten Tomatoes page of the director via Navigation, and then search for similar directors via Similar. After this, the users filled out a System Usability Scale questionnaire. We received a total of 13 responses, mostly from university students, giving a SUS score of 89.4, indicating very good *Usability*. However, more diverse and numerous user evaluations would be required in order to draw more robust conclusions: this should be considered a very preliminary (though encouraging) result. Users commented that certain visual aspects of the plugin could be improved, and some indicated a lack of familiarity with extensions (in general) and how to access them.

5. Discussion

We present DataQuest: a browser extension that uses Wikidata to augment Web navigation, providing users with several novel options to navigate to related entities within the same website, or to the pages of other websites about the same entity. Preliminary results suggest that DataQuest is responsive, usable, and has a low footprint, though more experiments would be necessary in future in order to draw more robust conclusions.

The main, current limitation of DataQuest is that it offers no functionality if the webpage is not found on Wikidata via an external identifier. While the features of DataQuest are not possible in such cases, other functionalities could be offered, such as enriching the website with semantic annotations of entities. Such annotations could be found (without ambiguity) via the links embedded in a webpage, matching them to external identifiers on Wikidata. An alternative would be to detect textual mentions of Wikidata entities on the webpage via entity linking, though this would require careful disambiguation. Upon hovering such links or entity mentions, a brief summary of the entity generated from Wikidata could be displayed.

Overall we see much potential in the idea of using Wikidata for Web augmentation, as we hope to have demonstrated with DataQuest.

Acknowledgments

This work was funded in part by ANID – Millennium Science Initiative Program – Code ICN17_002 and FONDECYT Regular 1221926. We also thank the anonymous reviewers whose feedback helped to improve the paper.

Declaration on Generative AI

Generative AI was not used in this research, nor in the preparation of this paper.

References

[1] N. O. Bouvin, Augmenting the Web through open hypermedia, New Rev. Hypermedia Multim. 8 (2002) 3–25. doi:10.1080/13614560208914733.

[2] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web, Scientific American 284 (2001).

[3] D. Vrandecic, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Commun. ACM 57 (2014) 78–85. doi:10.1145/2629489.

[4] I. Aldalur, Web augmentation: A systematic mapping study, Sci. Comput. Program. 232 (2024) 103045. doi:10.1016/J.SCICO.2023.103045.

[5] S. Firmenich, G. Rossi, M. Urbieta, S. E. Gordillo, C. Challiol, J. Nanard, M. Nanard, J. Araújo, Engineering Concern-Sensitive Navigation Structures, Concepts, Tools and Examples, J. Web Eng. 9 (2010) 157–185.

[6] C. Sottile, S. Firmenich, D. Torres, An End-User Semantic Web Augmentation Tool, in: End-User Development - 7th International Symposium, IS-EUD 2019, Hatfield, UK, July 10-12, 2019, Proceedings, volume 11553 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 239–243. doi:10.1007/978-3-030-24781-2_23.

[7] P. Oliveira, J. Rocha, Semantic annotation tools survey, in: IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013, Singapore, 16-19 April, 2013, IEEE, 2013, pp. 301–307. doi:10.1109/CIDM.2013.6597251.

[8] S. Corlosquet, R. Delbru, T. Clark, A. Polleres, S. Decker, Produce and Consume Linked Data with Drupal!, in: The Semantic Web - ISWC 2009, 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings, volume 5823 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 763–778. doi:10.1007/978-3-642-04930-9_48.

[9] V. Orabona, R. Palmieri, V. Moscato, A. Picariello, S. D'Elena, D. Cappetta, An application of semantic web technologies to enhance content management in web information portals, in: DATA 2015 - Proceedings of 4th International Conference on Data Management Technologies and Applications, Colmar, Alsace, France, 20-22 July, 2015, SciTePress, 2015, pp. 378–381. doi:10.5220/0005579803780381.

[10] D. Erenrich, Psychiq and Wwwyzzerdd: Wikidata completion using Wikipedia, Semantic Web Journal (2025). (Accepted / In press).

[11] S. Malyshev, M. Krötzsch, L. González, J. Gonsior, A. Bielefeldt, 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, October 8-12, 2018, Proceedings, Part II, volume 11137 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 376–394. doi:10.1007/978-3-030-00668-6_23.

[12] F. Å. Nielsen, Wembedder: Wikidata entity embedding web service, CoRR abs/1710.04099 (2017). arXiv:1710.04099.