A real-time visual dashboard for Wikidata edits

Damien Graux(✉), Fabrizio Orlandi(✉), Brian Lynch, Isobel Mahon, Odhran Mullen, Alex Mahon, Flora Molnar, and Lexes Mantiquilla

ADAPT SFI Research Centre & Trinity College Dublin, Ireland
{grauxd,orlandif}@tcd.ie

Abstract. During the last decades, the Web has seen the development of openly editable datasets on which users can suggest modifications at any moment. Recently, Wikidata as been the first large-scale Mediawiki-based dataset structured according to the Semantic Web standards. In this article, we propose the first version of a visual dashboard to allow real-time visualisation of Wikidata changes.

1 Introduction

Over the past two decades many data sources have been published on the Web. Most of the time, they follow the recommendations and standards promoted by the World Wide Web Consortium (W3C) within the Semantic Web movement, driven by the desire to create a "Web of data" from the conventional "Web of documents". These datasets, generally represented thanks to the RDF format [4] and accessible via the SPARQL language [7], deal with subjects ranging from generalist knowledge such as DBpedia [3], YAGO [5] or Wikidata [6] to specific knowledge such as legal court cases [1], source codes [2] or medical information [8]. Thus, the amount of semantic data now (publicly) accessible makes it possible to create new applications combining for instance several datasets at once.

Nevertheless, among the nowadays available datasets, multiple ones are actually open, meaning that users are able to contribute and pour new content directly into the knowledge base. This paradigm therefore allows each user to correct, amend, or refine the dataset. However, from a dataset maintainer perspective, such a feature increases the complexity of keeping track of the multiple data updates received. Practically, there exist various ways to follow changes of open data: from the history textual logs available for example on each Wikipedia page to the charts associated with each code-source repository of GitHub.

In particular, in 2014, Wikidata [6] –a collaboratively edited multilingual knowledge graph hosted by the Wikimedia Foundation– was released and it is a common source of open data that Wikimedia projects such as Wikipedia can use, and anyone else, under a public domain license. Practically, Wikidata currently contains 88 783 052 items and 1 258 940 393 edits have been made since

the project launch by at least 23 555 active users[1]. As a consequence, Wikidata is at the moment the largest collaboratively edited semantic knowledge base.

In this article, we describe the current efforts we are conducting to visually present the changes over Wikidata in (quasi) real time. The proposed interface, keeps track of the edits sent to Wikidata and updates our dashboard on-the-fly, letting users access to the latest status of the knowledge base.

2 Requirements and Technical Aspects

We extracted some technical requirements for the design of our application. The requirements elicitation process was performed having a particular use-case in mind. The end-user would be a Wikidata 'expert' who would like to monitor Wikidata edits in (quasi) real time in order to potentially identify anomalies, or discover interesting editing patterns (e.g. most active users and resources). The high-level requirements are:

- The data must be obtained from the Wikidata API.
- The visualisations displayed (charts and graphs) should be using the data collected from the API.
- The visualisations must be updated in quasi real time (a delay of a few seconds is acceptable).
- The user must be able to navigate through the web-app, select and expand different visualisations.
- The system should differentiate between edits performed by bots and humans.
- The system should display information about the most active users and resources (in edits volume).
- The type and time of each edit should be taken into account in the visualisations, along with contextual links pointing to the original edits on Wikidata.

A live web application has been selected as the most suitable form of presentation and interaction of the system. So to allow multiple online web users to experience our interface simultaneously. In order to deal with the real-time aspects of the application accordingly, we decided to use the ReactJS[2] framework, as a ready-to-go, well documented and widely used library. Using the endpoints from the Wikidata API, we created queries to search for all the relevant information in their database. Specifically, we wanted to observe the recent changes that are provided by the Mediawiki software[3]. The interface with the API was developed using pure JavaScript, without any additional libraries (e.g. jQuery). We then used HTML and CSS alongside the ReactJS framework to design a simple user interface. For the charts, we relied on the Nivo[4] JavaScript library, which provided us with React components to help with graphing data. This created very responsive and customisable graphs.

[1] From `https://www.wikidata.org/wiki/Wikidata:Statistics` (August 18th 2020)
[2] `https://reactjs.org/`
[3] `https://www.mediawiki.org/wiki/API:RecentChanges`
[4] `https://nivo.rocks/`

Fig. 1. Walk-through presenting the available graphic interfaces.

3 Wikidata Live Changes Web App

As shown in the application walk-through (see Figure 1), the user interface is made up of three parts. The homepage is the first page the user lands on and serves as a navigation hub providing the user with an array of options as to where to go next while also showing a few live statistics. From the homepage the user may choose between three buttons, the feed, the dashboard (Figure 2), or the user stats (a subcomponent of the dashboard). The feed allows the user to have a clear overview of the data coming in. The dashboard, the main part of the project, is where all the visualisations based on the incoming stream of data are located with each plot being interactive allowing for it to be made fullscreen or the data paused. Making a plot fullscreen gives the user information about

Fig. 2. The main page of the dashboard.

the plot they are looking at and adds labels to the plot, the user can also hover their mouse over a data point to see a preview for what said point represents.

More precisely, the dashboard (Figure 2), which is the central interface of the webapp, presents at a glance several visualisations: the most recent activity as a list of coming events, the recent edit size, the most active users, the most active pages, the largest recent edits and the proportion of edit flags. Moreover, each of these graphics is clickable, leading to a dedicated page providing more information. For instance Figure 3 presents details on the most recent edits: showing if the page has been freshly created or not, its size and who committed the changes. On a similar note, Figure 4 displays additional information on the most active users (whether they are human beings or bots) such as the size of the edits they made. Last but not least, the detailed interfaces also embed a "hovering" feature which allows to quickly glance inside sub-windows at some Wikidata resources (articles or user) without leaving the application.

Practically, it is important to note that the visualisations "start" when the user enters the page, meaning that the webapp does not keep track of the previously occurred events but rather begins "stacking" the edits made on Wikidata from the moment of connection. In addition, since there are often a dozen of changes per seconds, we included a *pause* functionality, in order to stop the application from displaying the coming changes in the interface. Once the pause button is pushed, the interface is frozen and the application keeps reading the edits in the back-end so that users would be served with the fresh data after releasing the pause.

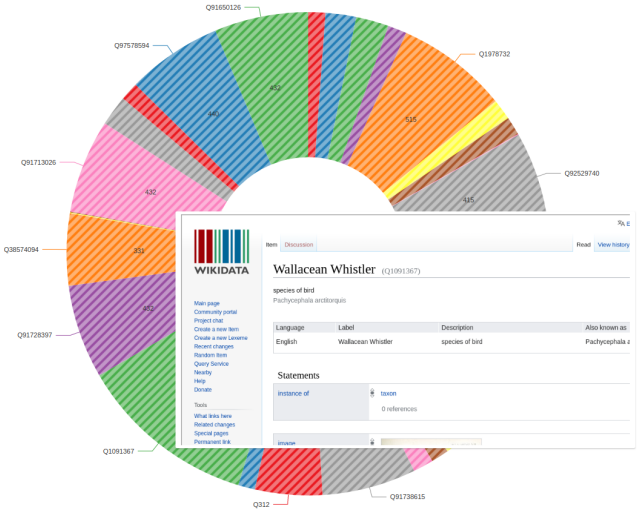

Fig. 3. A chart showing recent edits.

4 Conclusion

In this article, we described and shared our web-app to visualise Wikidata's edits in (quasi) real time . The presented interface is hosted on:

`https://isobelm.github.io/Software-Engineering/`

under an MIT license[5], providing users a live example of what the application could be locally, would someone be interested in deploying the interfaces at their premises. The data visualised by our website would allow researchers and Wikidata practitioners to easily identify anomalies or malicious edits to its databases.

We presented in this short article the first version of our live interface focused on Wikidata's edits. Practically, we are currently setting up a user validation experiment in order to improve the different snippets. On a different note, we are also planning to improve the webapp with additional features such as: allowing users to focus on specific Wikidata articles or letting users customize their dashboard. Moreover, we paid attention during the development not to restrict our architecture to the specific case of Wikidata, such that we can also add other data sources to our interfaces by adding calls to an additional API.

Acknowledgments

This research was conducted with the financial support of the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreements No. 801522 and No. 713567 at the ADAPT SFI Research Centre at Trinity College Dublin. The ADAPT SFI Centre for Digital

[5] Project's code base: `https://github.com/isobelm/Software-Engineering`

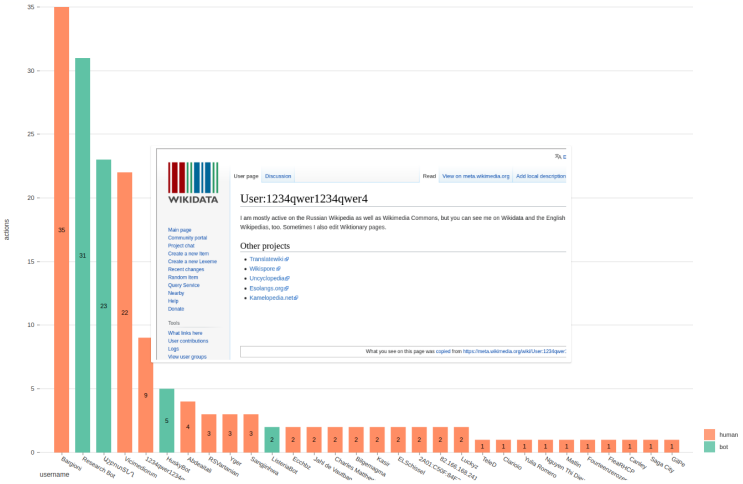

Fig. 4. Histogram of most active users.

Media Technology is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant #13/RC/2106.

References

1. Junior, A.C., Orlandi, F., Graux, D., Hossari, M., O'Sullivan, D., Hartz, C., Dirschl, C.: Knowledge graph-based legal search over german court cases. In: ESWC (2020)
2. Kubitza, D.O., Böckmann, M., Graux, D.: Semangit: A linked dataset from git. In: International Semantic Web Conference. pp. 215–228. Springer (2019)
3. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal **6**(2), 167–195 (2015), `http://jens-lehmann.org/files/2014/swj_dbpedia.pdf`
4. Manola, F., Miller, E., McBride, B., et al.: RDF primer. W3C recommendation **10**(1-107), 6 (2004)
5. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web. pp. 697–706. WWW'07, ACM, New York, NY, USA (2007). https://doi.org/10.1145/1242572.1242667
6. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)
7. W3C SPARQL Working Group, et al.: SPARQL 1.1 overview (2013), http://www.w3.org/TR/sparql11-overview/
8. Wishart, D.S., Knox, C., Guo, A.C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B., Hassanali, M.: Drugbank: a knowledgebase for drugs, drug actions and drug targets. Nucleic acids research **36**(suppl_1), D901–D906 (2008)