Preregistration: Comparing the use of Wikidata and Wikipedia by open-source software programmers on GitHub repositories

Houcemeddine Turki[1,*], Mohamed Ali Hadj Taieb[1], Mohamed Ben Aouicha[1], Lane Rasberry[2] and Daniel Mietchen[3,4]

[1]*Data Engineering and Semantics Research Unit, Faculty of Sciences of Sfax, University of Sfax, Sfax, Tunisia*
[2]*School of Data Science, University of Virginia, Charlottesville, VA, United States of America*
[3]*Ronin Institute for Independent Scholarship, Montclair, New Jersey, United States of America*
[4]*FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Berlin, Germany*

Abstract

Wikipedia and Wikidata are socio-technical systems driven by collaborative communities, open content, and open-source infrastructure. Some of the open-source software development around them involves code-sharing sites like GitHub. Analyzing GitHub for repositories related to Wikipedia and Wikidata can thus provide insights into multiple dimensions of the development of Wikipedia and Wikidata tools. We plan to do such an analysis, and in order to test our workflows for doing that, we ran a preliminary study based on a sample of 1000 GitHub repositories each for Wikidata and Wikipedia. We are preregistering our workflows here as a transparent basis for documenting and reporting on the full analysis later. The kinds of insights we expect based on the preliminary data about open-source GitHub repositories related to Wikidata and Wikipedia are as follows: (i) statistical information about these repositories; (ii) computational information, e.g. in terms of the programming languages used; (iii) demographic information about the contributors to such open-source projects; (iv) legal information about the choice of licenses; (v) linguistic information about the natural language used in the context of these repositories; (vi) trends over time. In the process of applying these preliminary workflows to studying the full dataset of GitHub repositories related to Wikipedia and Wikidata, we hope to gain some additional insights into the community dynamics at play in volunteer software development around Wikimedia projects, as well as into the process and merits of preregistrations for studies of this kind. We welcome community feedback on this approach as well as suggestions on additional aspects to include into the full study, and collaborations on the actual implementation.

Keywords

Wikipedia, Wikidata, Empirical Software Engineering, Wikimedia Developers, GitHub

Wikidata'23: Wikidata Workshop at ISWC 2023
*Corresponding author.
turkiabdelwaheb@hotmail.fr (H. Turki); mohamedali.hajtaieb@fss.usf.tn (M. A. Hadj Taieb); mohamed.benaouicha@fss.usf.tn (M. Ben Aouicha); lr2ua@virginia.edu (L. Rasberry); daniel.mietchen@ronininstitute.org (D. Mietchen)
0000-0003-3492-2014 (H. Turki); 0000-0002-2786-8913 (M. A. Hadj Taieb); 0000-0002-2277-5814 (M. Ben Aouicha); 0000-0002-9485-6146 (L. Rasberry); 0000-0001-9488-1870 (D. Mietchen)

1. Introduction

The field of *Empirical Software Engineering* has evolved around the idea and practice of studying how programmers work, e.g. how they address a particular issue or use a given resource to develop computer-based solutions [1]. With the age of Web 2.0, the creation of code repositories such as *GitHub*, *GitLab*, or *Gitee* has provided sufficient resources to study the behaviors of communities of computer programmers, particularly when they are contributing to collaborative projects [2]. As open-source projects have risen and provided successful outputs, the study of the characteristics of the communities behind them has grown significantly [3]. In this context, various patterns have been analyzed, ranging from the mechanisms behind the choice of topics [4], programming languages [5], and licenses [6] to the demographic distributions of communities [3, 7, 8]. Since the early days of the Wikimedia community, the development of open-source software has been one of the main pillars of the growth of Wikimedia projects, particularly *Wikipedia* and *Wikidata*. Open-source software within the realm of the Wikimedia community is diverse and involves, for instance, toolkits to process Wikipedia [9] and Wikidata [10], user scripts [11] and bots [12] to automate the editing and analysis of the two online databases [13], as well as tooling for visualizing Wikidata content [14], not to forget the software used to host Wikipedia and Wikidata, respectively – *MediaWiki* [15] and *Wikibase* [16].

This document represents a preregistration, outlining our approach to evaluating how programmers work on their development projects related to Wikipedia and Wikidata through the assessment of GitHub repositories about Wikidata and Wikipedia. Understanding the GitHub landscape of Wikipedia and Wikidata repositories is essential for optimizing collaboration, identifying programming trends, and ensuring the sustainability of these knowledge-sharing platforms. The primary research questions that guide our study include:

- How does the GitHub community interact with repositories related to Wikipedia and Wikidata?
- What are the predominant programming languages used in GitHub repositories related to Wikipedia and Wikidata?
- Who contributes the most to these GitHub repositories?
- What licenses are commonly associated with open-source projects related to Wikipedia and Wikidata?
- How is the number of repositories distributed by their year of creation and year of the last push?
- What are the most common topics of the repositories related to Wikipedia and Wikidata?

To address these questions, we will begin by explaining our approach to data collection and analysis based on a preliminary study using small-scale sample data and open-source code (Section 2). We will also highlight the importance of these research questions and their relevance in understanding the open-source landscape around Wikipedia and Wikidata. Following the data collection and analysis, we will provide preliminary results for this study based on the sample data and discuss them by contextualizing them with previous research findings on the matter (Section 3). Finally, we will draw conclusions on what the preliminary results mean for a larger-scale study and for potential future work targeting the open-source landscape

around Wikipedia and Wikidata (Section 4). Each figure and table presented on the way will contain a brief comment on how we expect the methodology of the full study to compare to the preliminary methodology presented here, emphasizing the significance of addressing the research questions to contribute to the understanding of software development in the Wikimedia community.

2. Methods

Table 1
Variables in the data retrieved for GitHub repositories. For the full study, we expect to use the same. *Category* is added to keep track of whether a repository is related to Wikipedia or Wikidata.

Variable	Definition	Type
Full name	The full name of the repository	object
Description	The short description of the repository	object
Home Page	The URL of the development project	object
Language	Main Programming Language of the repository	object
Number of forks	Number of forks from the repository	int64
Number of stars	Number of stars received by the repository	int64
License	License of the repository	object
User	GitHub Username of the Creator	object
Number of files	Number of files in the repository	int64
hasURL	Repository having a URL for the development project	bool
Year created	Year of the creation of the repository	object
Year of last push	Year of the last update of the repository	object
Category	Wikimedia Project related to the repository	object

For the initial study, we focus on *GitHub*, querying its search API via the dedicated Python library *PyGitHub* [17][1]. On April 20, 2023 we thus found 36.2k GitHub repositories related to Wikipedia[2] (Automatically assigned "Wikipedia" as a category) and 2.9k GitHub repositories related to Wikidata (Automatically assigned "Wikidata" as a category).[3] These ca. The GitHUb repositories initially included in both "Wikipedia" and "Wikidata" categories at once are attributed "Wiki" as a category. 40k repositories are the target of our full study aimed at analyzing the status of open-source software development related to the two Wikimedia Projects. In order to test our workflows, we retrieved – on the same day and still via PyGitHub – the metadata of the first 1,000 search results (that is the limit of the public search API) of the two search queries based on the *Best Match* sorting option (for the full study, we will retrieve the full dataset using date-based batches of 1000 or less).

[1]Code and data for our preliminary study are available via https://github.com/csisc/WikiGitHub.
[2]https://github.com/search?q=wikipedia&type=repositories
[3]https://github.com/search?q=wikidata&type=repositories

2.1. Structure of the data

The retrieved data involves multiple variables about the characteristics of repositories and their activity, as shown in Table 1. We stored the dataset as an Excel spreadsheet using *Pandas*, a Python Library for data analytics [18]. Later, we used *Matplotlib* and *Seaborn* [19], two Python Libraries for data visualization, to generate plots to visualize the statistical features of the variables. None of the retrieved data except the *Description* field (see Section 3.6 for details) and the *License* field (see Section 3.4 for details) have been pre-processed before data analysis.

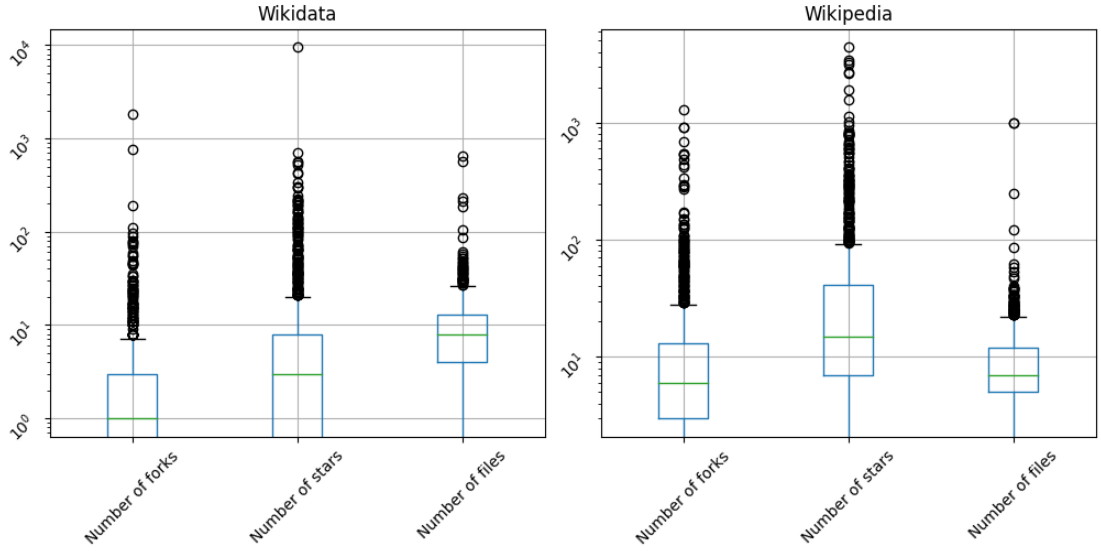

Figure 1: Boxplot for the number of stars, forks, and files for Wikidata and Wikipedia repositories. For the full study, we expect several effects to be relevant. (A) The number of repositories included in the graph will grow to about 3-fold for Wikidata and about 37-fold for Wikipedia. We expect this to lead to smaller variances and that this effect should be much more pronounced for the Wikipedia data. (B) As existing repositories age, the values along the three dimensions plotted here are expected to all rise as a consequence. (C) New repositories will be created, which will lower those values.

3. Preliminary results and discussion

3.1. Statistical information about the repositories

Of the 2000 *GitHub* repositories retrieved in total, some (18 for Wikipedia and 6 for Wikidata) were empty. Since that does not provide information about software development, we eliminated them in favour of a more meaningful analysis, leaving 1,976 non-blank GitHub repositories related to either or both of *Wikipedia* or *Wikidata*. 982 of them are exclusively related to Wikipedia, while 994 are related to Wikidata[4]. This distribution allows a fair comparison

[4]Wikidata-related repositories include 35 generic repositories also linked to Wikipedia. We disregard this fact for the purpose of our preliminary study. For the full study, we expect this number to be higher, and plan to explore the connections.

between the patterns in which Wikipedia and Wikidata are used in computer programming, whereas the full study will mean a comparison based on unequal numbers (somewhat higher than the 36.2k for Wikipedia versus 2.9k for Wikidata that were reported above). 78.3% of the Wikidata-related repositories and 68.6% of the Wikipedia-related ones returned a value for the *Home Page* URL, so presumably have a project website. This can be interpreted to mean that the Wikidata-related repositories and to a lesser extent the Wikipedia-related repositories are developed for practical use in deploying web tools and services [20].

When assessing the popularity and volume of the GitHub repositories, we found that the average number of stars and forks for Wikipedia-related repositories significantly exceeds the ones for Wikidata-related repositories, as shown in Figure 1. This is mainly due to the higher popularity and longer age of Wikipedia. *Wikipedia* has been created in 2001, so it is 11 years older than *Wikidata* [10]. In terms of audience, *Wikipedia* is used by millions of people across the world for information seeking in various contexts, making it one of the most visited websites for years [21]. By contrast, despite being multilingual, the direct readership of *Wikidata* is only equal to 2% of that of the English Wikipedia, as of June 2023 (277 million vs. 10 billion pageviews)[5]. Thanks to their openly licensed content, both projects are used indirectly too, e.g. on other websites, in search engines, or in research projects. Here, we are not aware of comparative data, but we suspect that such indirect uses would see Wikidata at least not far behind Wikipedia, and possibly ahead of it.

3.2. Computational information

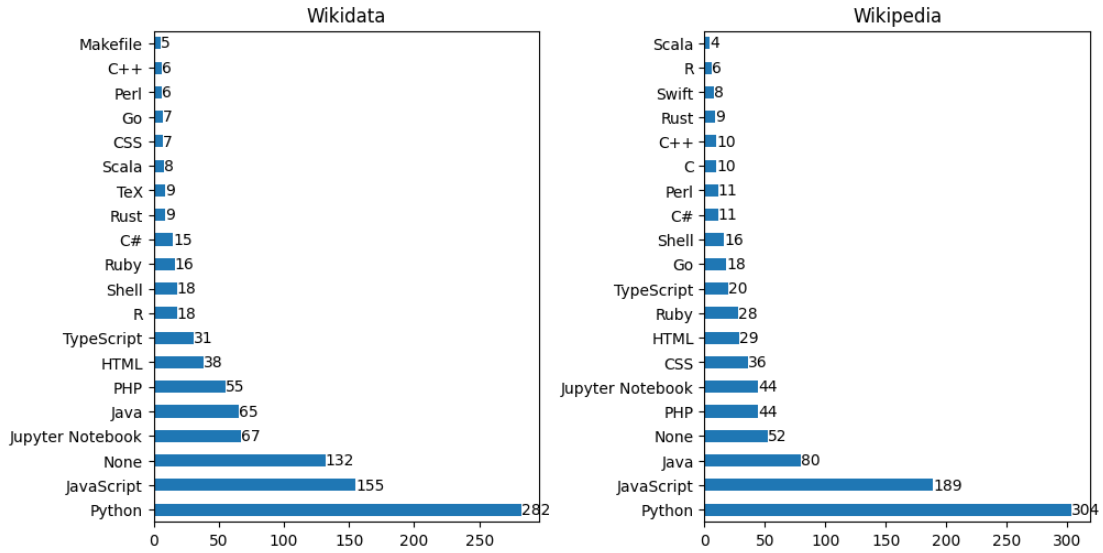

Figure 2: Wikidata and Wikipedia GitHub repositories by main programming language. "None" refers to repositories that primarily hold data rather than code. For the full study, we expect a similar picture, perhaps with a higher visibility of currently popular languages.

[5]Live data: https://stats.wikimedia.org/#/wikidata.org and https://stats.wikimedia.org/#/en.wikipedia.org.

When examining the main programming language for GitHub repositories related to Wikidata and Wikipedia, we found a similar profile about the choice of programming languages to process Wikidata and Wikipedia, as shown in Figure 2, despite the fact that Wikipedia is semi-structured, while Wikidata is a fully-structured open knowledge graph [10]. In both cases, we found the distribution of GitHub repositories per main programming language to follow a Lotka-like law (i.e., an inverse power law) [22]. This is concordant with previous research findings on the matter [23]. Overall, *Python* was clearly the most popular language in the repositories we explored (it was used in 282 Wikidata-related repositories and 304 Wikipedia-related repositories). This preponderance of Python is a pattern common to research-oriented GitHub repositories [23]. It is not, however, a general characteristic of GitHub repositories, where *JavaScript*, a web programming language, is leading the worldwide open-source development movement [24, 25]. The prominence of Python can be explained not only by the general popularity of this language [26] but also (somewhat relatedly) by the better availability of robust Python libraries that are customized to process Wikimedia projects. These include *Pywikibot* (https://pypi.org/project/pywikibot/), *Wikibase Integrator* (https://pypi.org/project/wikibaseintegrator/), *WPTools* (https://pypi.org/project/wptools/), and *Wikipedia* (https://pypi.org/project/wikipedia/). That being said, web programming languages such as *JavaScript*, *CSS*, *PHP*, and *HTML* are also popular in Wikidata and Wikipedia open-source development projects. This is mainly due to the fact that *Wikidata* and *Wikipedia* are web-hosted projects [10]. *Java* appeared among the most used programming languages too. This can be understood in terms of the common use of this object-oriented programming language for open-source development projects [24, 25] and Java's popularity amongst researchers [27] as well as for mobile app development [28]. Jupyter notebooks were also popular, and while we did not analyze the language they were written in, other analyses suggest that they are mostly written in Python, though increasingly in other languages too, like R, Julia, and Scala [29].

3.3. Demographic information

When retrieving the GitHub usernames of the accounts behind the repositories, we found the repositories to be distributed among developers in a way that follows Lotka's law [22], as shown in Figure 3. This fits with the overall patterns of the distribution of GitHub repositories per author [23]. Most prominent in the data is the *Wikimedia Foundation* Development Team (17 Wikipedia-related repositories and 14 Wikidata-related ones). Although their GitHub repositories are generally mirrors of the development happening on Wikimedia servers, this team's prominence in our results highlights its role in maintaining the software behind Wikimedia projects, particularly *MediaWiki* and *Pywikibot* [30]. *Wikimedia Deutschland*, the Wikimedia chapter in Germany, created the largest number of GitHub repositories related to Wikidata (17). This confirms the central position of *Wikimedia Deutschland* in advocating for, promoting, and developing *Wikidata* as a project that can bring other Wikimedia projects to the next stage [31]. The analysis of the other main contributors to Wikidata and Wikipedia GitHub repositories revealed that most of the contributions of Wikipedia-related repositories are either tech giants like *Google*, startups and open-source communities like *hatnote*, *Kiwix*, *learn-co-students*, and *Wikifox*, or research scientists (e.g. *Ed Summers* [Stanford University, United States of America] and *Brian C. Keegan* [University of Colorado Boulder, United States of America]) and projects

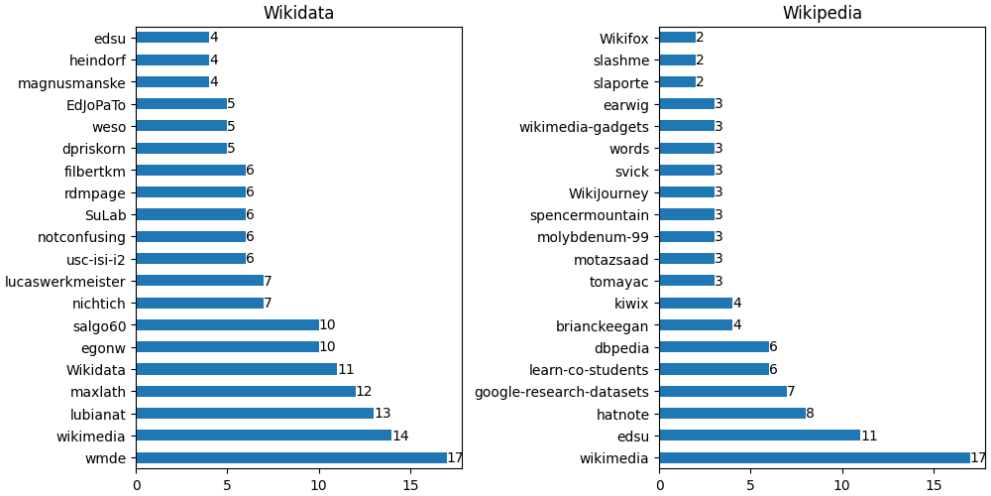

Figure 3: Frequent creators of Wikidata and Wikipedia repositories on GitHub. For the full study, we expect a similar picture, with changes concentrated on the tail end rather than the lead.

(e.g., *DBpedia*). We identified no developer who started more than three Wikipedia-related repositories.

The situation is different for Wikidata, where we observed Wikimedia volunteers who had established more than three relevant repositories, including Tiago Lubiana (*lubianat*) and Maxime Lathuilière (*maxlath*) at 13 and 12, respectively. Wikimedia researchers having a close relationship with the Wikimedia Community are also visibly engaged in Wikidata open-source development (e.g., Egon Willighagen [*egonw*], Andrew I. Su [*sulab*], and Jakob Voß [*nichtich*]). The higher involvement of the Wikimedia Community in developing Wikidata-related projects can be explained in part by the more standardized format of Wikidata in the form of triples, which makes it easier than Wikipedia to process automatically [10]. Other factors likely contributing to the effect include the fact that Wikidata tools can to some extent build on Wikipedia ones, and the considerable efforts that *Wikimedia Deutschland* is investing in the disseminating of technical aspects of Wikidata, particularly inside the community [31]. Most of the main contributors to GitHub repositories related to Wikidata and Wikipedia are from Europe and North America. This is mainly caused by the country distribution of the open-source development community [7] and fits with similar analyses of GitHub contributor demographics, including one that combined GitHub data about repositories with demographic data from Wikidata [8]. All of the active contributors we named above self-identify as male. Gender bias is a common feature of the GitHub open-source community [8], with various contributing factors [32, 33, 34]. Further efforts should be provided to promote diversity, equity, and inclusion inside the Wikimedia technical community. Works in this direction have already been provided to establish a more inclusive community of Wikidata contributors, enhancing the coverage of under-represented topics in the knowledge graph [35].

3.4. Legal information

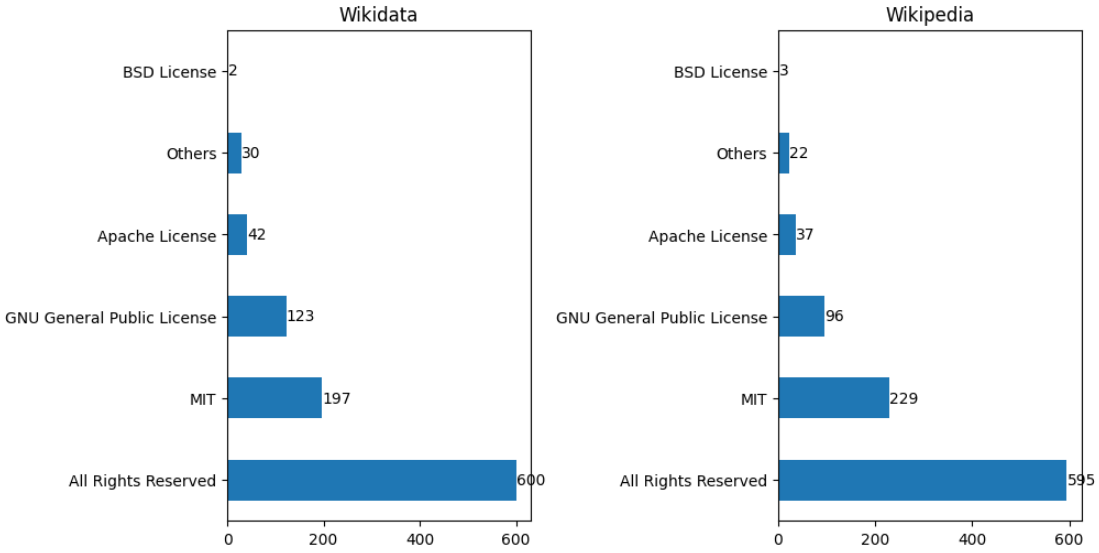

Figure 4: Wikidata and Wikipedia GitHub repositories by software license. "All Rights Reserved" essentially means the absence of a license declaration. For the full study, we expect this pattern to hold.

Among the information that have been retrieved using the GitHub API, there was data about license attribution based on the first line of every license file. These first lines of license files are processed by hand to attribute the right license name for every repository. The analysis of the licenses assigned to the GitHub repositories related to Wikidata and Wikipedia finds both to have a similar profile of license attribution, as shown in Figure 4. The apparent dominance of *All Rights Reserved* might come as a surprise but has to be understood such that those repositories did not have any license declared, in which case full copyright protection has to be assumed by default. Further efforts should be aimed at raising awareness within the Wikimedia technical community around the importance of using – and properly declaring – permissive licenses to allow the reuse and upgrade of their source codes for the good of the Wikimedia projects.

The most popular open license in Wikimedia open-source development is the *MIT License*, followed by the *GNU General Public License* and the *Apache License*. These three licenses are the main ones that have been used for years in open-source development on GitHub [36, 37]. The only surprising fact is that the rate of *GNU General Public License* use is higher in Wikidata-related repositories in the same way as the rate of *MIT License* is higher in Wikipedia-related repositories. The reduced gap between *GNU General Public License* use and *MIT License* use in Wikidata-related repositories can perhaps be explained by the fact that Wikidata is released under the CC0 License [10] and that Wikipedia is released under the CC-BY-SA 4.0 License [38]. The GNU General Public License is more compatible with the CC0 License than the CC-BY-SA 4.0 License [39].

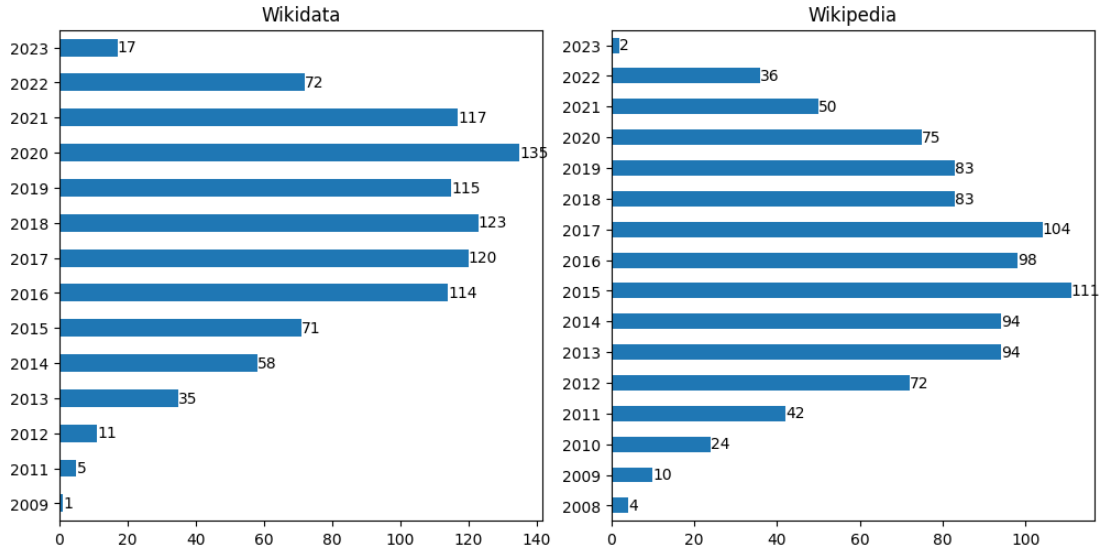

Figure 5: GitHub repositories related to Wikidata and Wikipedia by creation year. This is hard to interpret other than that it likely reflects not only the underlying data but also some strong biases. While the low values for 2023 are an artifact of having queried early in the year, we believe that this figure contains further artifacts – some of which correlate with repository age – that were introduced by sampling only 1000 repositories each for both websites. For the full study, we expect the values for recent years to be higher relative to the maximum.

3.5. Trends

Changes over time are difficult to assess with the limited dataset that we have, since we suspect that the sort order we used to sample GitHub repositories is influenced by at least some parameters that correlate with repository age, e.g. the number of stars, forks, commits, commiters, or traffic.

In particular, the sharp decline in recent years that is visible in Figure 5 might well be an artifact. This could have been introduced due to our sampling a limited amount of repositories in a non-random fashion, or it could be due to more subtle effects, such as the time that it takes before a GitHub-hosted repository is actually indexed by GitHub and included into its search index.

We thus refrain from interpreting the distribution given in Figure 5 based on the preliminary data, except for noting the existence of a few repositories that deal with Wikidata and that have been created before the inception of the Project in 2012. These projects are either preliminary sources for the initial development of Wikidata [31] or several Wikipedia-related development projects that have been adapted to support Wikidata as a resource like YAGO [40], an open knowledge graph initially derived from Wikipedia and WordNet [40].

Notwithstanding the suspected bias against new repositories, recent years feature strongly in terms of the year of last commit, and 2023 – despite being in its early stages at the time of sampling – came out on top. Once we have the full dataset, we expect the prominence of recent years in this plot to be even more pronounced, and comparisons to the temporal

patterns of repository creation (as per Figure 5) might yield insights into community dynamics, sustainability and related matters. There could also be relationships between the license choice (cf. Figure 4) and commit trends, since more permissive licenses provide more avenues for engagement with a given repository [41].

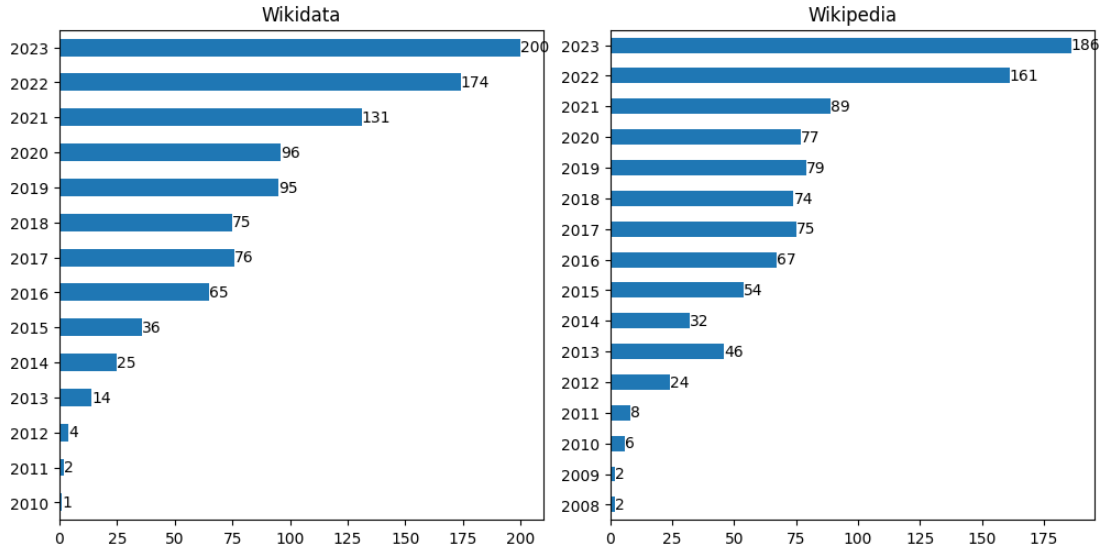

Figure 6: GitHub repositories related to Wikidata and Wikipedia by year of last public commit. For the full study, we expect the recent years to be even more prominent.

3.6. Linguistic information

The *Description* data obtained from GitHub for each repository (cf. Table 1 - Brief statements available for all the considered repositories) have been converted to lowercase, split by spaces, and stripped from stopwords and punctuation to identify the most common words provided to describe the GitHub repositories related to Wikidata and Wikipedia (cf. Fig. 7).

The analysis of the top words included in these descriptions revealed that the distribution of the words in the descriptions follows Zipf's law [42].

Both sets of GitHub repositories mainly deal with applications for the processing of data of the two projects, as highlighted by the prominence of the word *data* in both. This involves the management of Wikipedia pages (*articles*, *pages*, *text*, *article*, and *page*) and Wikidata items and statements (*knowledge*, *items*, *graph*, and *entities*). This also includes the development of software (*code*), software libraries (*library*) – especially in *python*) – for the development of Application Programming Interfaces (APIs) and dump processing methods (*api*, *wikibase*, *mediawiki*, *dumps*, and *dump*) and the creation of tools (*tool* and *tools*) and projects (*project*). This is mostly done in the context of promoting the systematic use of APIs and dumps for the automatic enrichment and processing of Wikidata and Wikipedia [43].

Beyond this, there are several specific applications that are only applicable to Wikidata or Wikipedia. Wikidata repositories emphasize projects related to the management of SPARQL

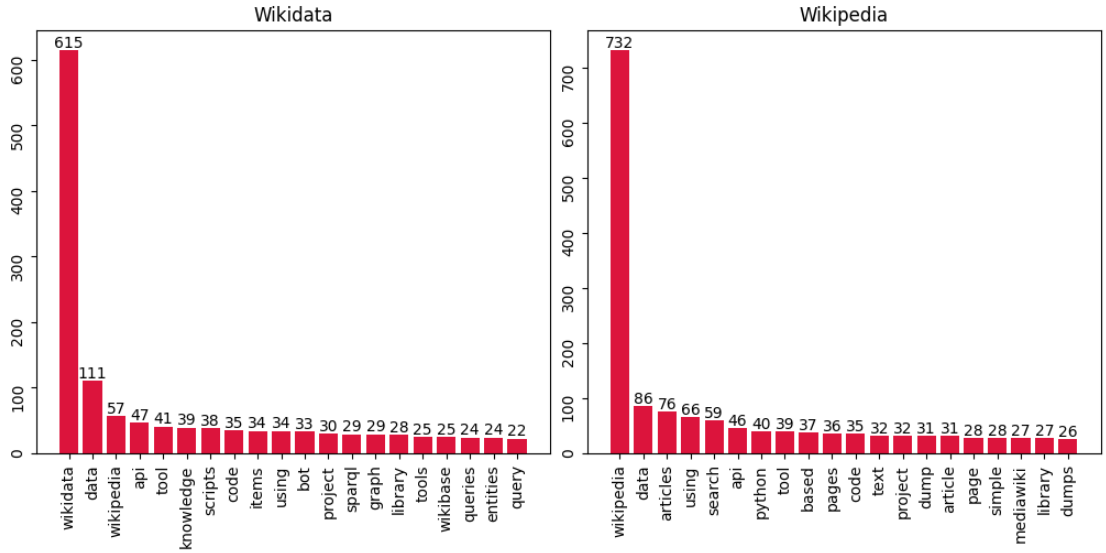

Figure 7: Top words in the descriptions of GitHub repositories related to Wikidata (left) or Wikipedia (right). Stopwords have been removed, no stemming has been performed. All data shown throughout the manuscript is based on a sample of 1000 repositories for each of the two websites. For the full study, we expect the top ranks to be rather stable and some reshuffling below them, perhaps with some terms appearing that are missing in the current results – possible candidates for this could be names of tool-related platforms like Toolforge or PAWS, or of commonly used services like OAuth.

queries (*sparql*, *queries*, and *query*) or the creation of bots or scripts for the automatic enrichment of the knowledge graph. This is closely linked to ongoing research projects for Wikidata related to SPARQL query optimization and data augmentation based on external resources [13].

As for Wikipedia repositories, they show an interest in the development of methods for quicker and simpler data mining of the project (*search* and *simple*). This confirms that the development efforts around Wikipedia meet the long-term research efforts for the development of robust and more efficient data mining techniques for exploring Wikipedia [44].

4. Conclusion

This project lightly analyzed about 2000 repositories out of nearly 40,000 that exist on GitHub. Now that this project exists as a demonstration of how the analysis could work, we could gain higher precision in our findings and also expand to more easily identify exceptional cases if we continued to analyze all identified repositories. Furthermore, we could include GitLab or Codeberg repositories – which have a reputation among some Wikimedia developers for being more value-aligned than GitHub – or repositories from other platforms like Gitee that have other demographic biases. We could also take a look at the technical communication platforms of the Wikimedia Foundation such as *Gerrit*[6] and *Phabricator*[7] [45] to further examine how

[6]https://gerrit.wikimedia.org/r/.
[7]https://phabricator.wikimedia.org.

the Wikimedia technical community discusses the incremental development of collaborative projects, possibly including hardware-related ones like *Internet-in-a-Box*[8].

This project explored which questions would be answerable with the data available in GitHub. More insights could be gained from matching external datasets to this data (e.g. as per [8]), including the disambiguation of contributors and institutions, matching repositories to scholarly publications [46, 47], and usage statistics in Wikimedia platforms.

The Wikimedia community is highly engaged in the governance of Wikimedia projects. This engagement plays out in various ways, but for example, community forums exist in the Wikimedia platform where developers and users meet to discuss user challenges and technical possibilities. While the Wikimedia community will discuss this paper as they routinely do for all such reports, some people will undoubtedly ask further questions, and others may want to interpret the preliminary results from this paper to inform ongoing or planned tool development. We intended for this first analysis to be useful, but given the long-term budget planning of the Wikimedia Foundation for development, the global and large Wikimedia audience base, and the stakes of sustaining success as a nonprofit general information resource, scheduled reporting updates for development trends such as these would surely guide stakeholder decision making. One notable concern highlighted by our analysis is the prevalence of "All Rights Reserved" licenses in repositories. This finding underscores the importance of educating developers on open science best practices and encouraging them to declare a license for their projects, even if they are not open-source in nature. This step can contribute to a more open and collaborative development ecosystem within Wikimedia and similar communities. This preregistration is only an early step into characterizing the development landscape. Correct collection and interpretation of development statistics such as those we explored here – especially considering how much of this is volunteer-organized with little central planning – could have significant returns of community engagement on the investment.

While Wikipedia and Wikidata have been success stories in many respects – including some software-related ones –, we suspect that volunteer developers of software around Wikipedia, Wikidata and other Wikimedia projects would still benefit from insights into the social dynamics of software development (within Wikimedia contexts as well as more generally), and so we invite feedback from anyone who might be a potential user of the results of the full study.

Acknowledgments

This research is funded by the Wikimedia Research Fund of Wikimedia Foundation (San Francisco, California, United States of America) through the *Adapting Wikidata to support clinical practice using Data Science, Semantic Web and Machine Learning* Project.[9] Source code and data are made available under the MIT License at https://github.com/csisc/WikiGitHub.

[8] https://meta.wikimedia.org/wiki/Internet-in-a-Box

[9] https://meta.wikimedia.org/wiki/Research:Adapting_Wikidata_to_support_clinical_practice_using_Data_Science,_Semantic_Web_and_Machine_Learning

References

[1] D. M. Fernández, J.-H. Passoth, Empirical software engineering: From discipline to interdiscipline, Journal of Systems and Software 148 (2019) 170–179. doi:10.1016/j.jss.2018.11.019.

[2] G. Zhao, D. A. da Costa, Y. Zou, Improving the pull requests review process using learning-to-rank algorithms, Empirical Software Engineering 24 (2019) 2140–2170. doi:10.1007/s10664-019-09696-8.

[3] A. Bosu, K. Z. Sultana, Diversity and Inclusion in Open Source Software (OSS) Projects: Where Do We Stand?, in: 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE, 2019. doi:10.1109/esem.2019.8870179.

[4] A. Hindle, C. Bird, T. Zimmermann, N. Nagappan, Do topics make sense to managers and developers?, Empirical Software Engineering 20 (2014) 479–515. doi:10.1007/s10664-014-9312-1.

[5] B. Ray, D. Posnett, V. Filkov, P. Devanbu, A large scale study of programming languages and code quality in GitHub, in: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, ACM, 2014. doi:10.1145/2635868.2635922.

[6] C. Vendome, A Large Scale Study of License Usage on GitHub, in: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, IEEE, 2015. doi:10.1109/icse.2015.245.

[7] J. Wachs, M. Nitecki, W. Schueller, A. Polleres, The Geography of Open Source Software: Evidence from GitHub, Technological Forecasting and Social Change 176 (2022) 121478. doi:10.1016/j.techfore.2022.121478.

[8] E. Levitskaya, G. Korkmaz, D. Mietchen, L. Rasberry, Analysis of linked github and wikidata, 2022. doi:10.5281/zenodo.7443339, The Alfred P. Sloan Foundation supported this project with grant G-2021-17106.

[9] D. Milne, I. H. Witten, An open-source toolkit for mining Wikipedia, Artificial Intelligence 194 (2013) 222–239. doi:10.1016/j.artint.2012.06.007.

[10] D. Vrandečić, M. Krötzsch, Wikidata, Communications of the ACM 57 (2014) 78–85. doi:10.1145/2629489.

[11] H. Turki, M. A. Hadj Taieb, M. Ben Aouicha, Coupling Wikipedia Categories with Wikidata Statements for Better Semantics, in: 2nd Wikidata Workshop (Wikidata@ISWC 2021), CEUR Workshop Proceedings, 2021, p. 8. URL: https://ceur-ws.org/Vol-2982/paper-8.pdf.

[12] E. Seidlmayer, J. Voß, T. Melnychuk, L. Galke, K. Tochtermann, C. Schultz, K. U. Förstner, ORCID for Wikidata - Data enrichment for scientometric applications, in: 1st Wikidata Workshop (Wikidata@ISWC 2020), CEUR Workshop Proceedings, 2020, p. 9. URL: https://ceur-ws.org/Vol-2773/paper-09.pdf.

[13] M. Farda-Sarbas, C. Müller-Birn, Wikidata from a Research Perspective – A Systematic Mapping Study of Wikidata, 2019. doi:10.48550/ARXIV.1908.11153.

[14] F. Å. Nielsen, D. Mietchen, E. Willighagen, Scholia, Scientometrics and Wikidata, in: The Semantic Web: ESWC 2017 Satellite Events, 2017, pp. 237–259. doi:10.1007/978-3-319-70407-4_36.

[15] Y. Koren, Working with MediaWiki, WikiWorks Press, San Bernardino, CA, USA, 2012.

[16] L. Rossenova, P. Duchesne, I. Blümel, Wikidata and Wikibase as complementary research data management services for cultural heritage data, in: 3rd Wikidata Workshop (Wikidata@ISWC 2022), CEUR Workshop Proceedings, 2022, p. 15. URL: https://ceur-ws.org/Vol-3262/paper15.pdf.

[17] V. Jacques, Pygithub: Typed interactions with the github api v3, 2021. URL: https://github.com/PyGithub/PyGithub.

[18] W. McKinney, pandas: a foundational Python library for data analysis and statistics, Python for High Performance and Scientific Computing 14 (2011) 1–9.

[19] E. Bisong, Matplotlib and seaborn, in: Building Machine Learning and Deep Learning Models on Google Cloud Platform, Apress, 2019, pp. 151–165. doi:10.1007/978-1-4842-4470-8_12.

[20] T. Puhlfurs, L. Montgomery, W. Maalej, An Exploratory Study of Documentation Strategies for Product Features in Popular GitHub Projects, in: 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME), IEEE, 2022. doi:10.1109/icsme55016.2022.00043.

[21] C. Okoli, M. Mehdi, M. Mesgari, F. Å. Nielsen, A. Lanamäki, Wikipedia in the eyes of its beholders: A systematic review of scholarly research on Wikipedia readers and readership, Journal of the Association for Information Science and Technology 65 (2014) 2381–2403. doi:10.1002/asi.23162.

[22] M. L. Pao, Lotka's law: A testing procedure, Information Processing & Management 21 (1985) 305–320. doi:10.1016/0306-4573(85)90055-x.

[23] M. Färber, Analyzing the GitHub Repositories of Research Papers, in: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020, ACM, 2020. doi:10.1145/3383583.3398578.

[24] B. Ray, D. Posnett, P. Devanbu, V. Filkov, A large-scale study of programming languages and code quality in GitHub, Communications of the ACM 60 (2017) 91–100. doi:10.1145/3126905.

[25] D. Celińska, E. Kopczyński, Programming Languages in GitHub: A Visualization in Hyperbolic Plane, Proceedings of the International AAAI Conference on Web and Social Media 11 (2017) 727–728. doi:10.1609/icwsm.v11i1.14862.

[26] T. F. Bissyande, F. Thung, D. Lo, L. Jiang, L. Reveillere, Popularity, Interoperability, and Impact of Programming Languages in 100,000 Open Source Projects, in: 2013 IEEE 37th Annual Computer Software and Applications Conference, IEEE, 2013. doi:10.1109/compsac.2013.55.

[27] S. Wattanakriengkrai, B. Chinthanet, H. Hata, R. G. Kula, C. Treude, J. Guo, K. Matsumoto, Github repositories with links to academic papers: Public access, traceability, and evolution, Journal of Systems and Software 183 (2022) 111117. doi:10.1016/J.JSS.2021.111117.

[28] I. J. Mojica, B. Adams, M. Nagappan, S. Dienst, T. Berger, A. E. Hassan, A Large-Scale Empirical Study on Software Reuse in Mobile Apps, IEEE Software 31 (2014) 78–86. doi:10.1109/ms.2013.142.

[29] S. Samuel, D. Mietchen, Computational reproducibility of jupyter notebooks from biomedical publications, 2022. arXiv:2209.04308.

[30] B. Lin, G. Robles, A. Serebrenik, Developer turnover in global, industrial open source projects: Insights from applying survival analysis, in: 2017 IEEE 12th International

Conference on Global Software Engineering (ICGSE), IEEE, 2017. doi:10.1109/icgse.2017.11.

[31] D. Vrandečić, L. Pintscher, M. Krötzsch, Wikidata: The Making Of, in: Companion Proceedings of the ACM Web Conference 2023, ACM, 2023. doi:10.1145/3543873.3585579.

[32] Z. Wang, Y. Wang, D. Redmiles, Competence-confidence gap: a threat to female developers' contribution on github, in: Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Society, ACM, 2018. doi:10.1145/3183428.3183437.

[33] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, V. Filkov, Gender and Tenure Diversity in GitHub Teams, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, ACM, 2015. doi:10.1145/2702123.2702549.

[34] J. Terrell, A. Kofink, J. Middleton, C. Rainear, E. Murphy-Hill, C. Parnin, J. Stallings, Gender differences and bias in open source: pull request acceptance of women versus men, PeerJ Computer Science 3 (2017) e111.

[35] S. Allison-Cassin, A. Armstrong, P. Ayers, T. Cramer, M. Custer, M. Lemus-Rojas, S. McCallum, M. Proffitt, M. A. Puente, J. Ruttenberg, A. Stinson, ARL White Paper on Wikidata: Opportunities and Recommendations, Association of Research Libraries, Washington, DC, 2019. URL: https://www.arl.org/wp-content/uploads/2019/04/2019.04.18-ARL-white-paper-on-Wikidata.pdf.

[36] C. Vendome, G. Bavota, M. D. Penta, M. Linares-Vásquez, D. German, D. Poshyvanyk, License usage and changes: a large-scale study on GitHub, Empirical Software Engineering 22 (2016) 1537–1577. doi:10.1007/s10664-016-9438-4.

[37] X. Wu, J.-Y. Wu, M.-H. Zhou, Z.-Q. Wang, L.-Y. Yang, Analysis of open source license selection for the GitHub programming community, 2020. doi:10.48550/ARXIV.2009.00981.

[38] J. M. Heilman, E. Kemmann, M. Bonert, A. Chatterjee, B. Ragar, G. M. Beards, D. J. Iberri, M. Harvey, B. Thomas, W. Stomp, M. F. Martone, D. J. Lodge, A. Vondracek, J. F. de Wolff, C. Liber, S. C. Grover, T. J. Vickers, B. Meskó, M. R. Laurent, Wikipedia: A Key Tool for Global Public Health Promotion, Journal of Medical Internet Research 13 (2011) e14. doi:10.2196/jmir.1589.

[39] G. Hagedorn, D. Mietchen, R. Morris, D. Agosti, L. Penev, W. Berendsohn, D. Hobern, Creative Commons licenses and the non-commercial condition: Implications for the re-use of biodiversity information, ZooKeys 150 (2011) 127–149. doi:10.3897/zookeys.150.2189.

[40] T. P. Tanon, G. Weikum, F. Suchanek, YAGO 4: A Reason-able Knowledge Base, in: The Semantic Web, Springer International Publishing, 2020, pp. 583–596. doi:10.1007/978-3-030-49461-2_34.

[41] S. Rathee, A. Chobe, Open source growth and trends, in: Getting Started with Open Source Technologies, Apress, 2022, pp. 149–169. doi:10.1007/978-1-4842-8127-7_8.

[42] M. Newman, Power laws, Pareto distributions and Zipf's law, Contemporary Physics 46 (2005) 323–351. doi:10.1080/00107510500052444.

[43] T. Steiner, Bots vs. Wikipedians, Anons vs. Logged-Ins (Redux), in: Proceedings of The International Symposium on Open Collaboration, ACM, 2014. doi:10.1145/2641580.2641613.

[44] Y. Wang, J. Zhang, Exploring topics related to data mining on Wikipedia, The Electronic Library 35 (2017) 667–688. doi:10.1108/el-09-2016-0188.
[45] W. Brown, How is the speed of code review affected by activity, usage and code quality?, 2023. doi:10.48550/ARXIV.2305.05770.
[46] L. Rasberry, D. Mietchen, Scholia for Software, Research Ideas and Outcomes 8 (2022) e94771. doi:10.3897/RIO.8.E94771.
[47] A.-M. Istrate, D. Li, D. Taraborelli, M. Torkar, B. Veytsman, I. Williams, A large dataset of software mentions in the biomedical literature (2022). doi:10.48550/ARXIV.2209.00693.