Deyuan Yang
08/02/2023
Final report, Distributed Research Experiences for Undergraduates

"WhoFundedIt" Web Application

Abstract

A funder is a person or an organization that provides money to pay for research. Funders are essential for research, but sometimes research has misleading outcomes due to a funder's interest. This project aims to design and develop the "WhoFundedIt" web application to visualize the funder information for each research project in a list of scientific publications. First, I designed the minimum viable product, the simplest version of the application. Then, I identified four technology stacks and analyzed their pros and cons. I implemented the technology stack based on the Shiny framework. The input for my "WhoFundedIt" App prototype is publications' DOIs, and its output is funder data from Crossref. In the future, I will add additional funder data sources, visualizations, and more input and output formats options. The "WhoFundedIt" web application will be a useful tool to help with funder transparency and identifying potential sources of bias. GitHub link to the web application: https://github.com/infoqualitylab/WhoFundedIt-app

Keywords: funder, funding bias, funding transparency, web application, bias identification

1. Introduction

Research funding plays a crucial role in supporting scientific research and advance knowledge in various areas. It is important for researchers secure appropriate funding resources, but sometimes it is challenging. It is necessary to have a reliable tool to help researcher analyze potential funders who are more likely to support their projects with the data about their previous funded projects. In addition, funders can determine which research can be conducted to impact people's perception (Krimsky, 2013). However, it can also bring biases and conflicts of interest into research outcomes due to the tendency of the scientific study to support the interests of the financial sponsor (Kee, 2021). It is important to understand the funding sources of scientific publications to promote funding transparency in the research. The "WhoFundedIt" web application aims to address the demands for funder transparency and identify potential funders for scientific research.

The goal of this paper is to describe the design and development of process of the "WhoFundedIt" web application. The paper will introduce the minimum viable product, which is the simplest version of the program with accuracy evaluation, four technology stacks to design

and develop the web application, and also the implementation of the chosen technology stack with deep analysis of each design, functionality and feature. Besides, the paper will discuss the limitations of designing and developing “WhoFundedIt” web application, as well as future improvements for it. Furthermore, the paper explores the potential impact of the web application in the field of research, especially in research transparency and the identification of potential sources of bias. By easily accessing the funder information for scientific publications, people can identify funding opportunities effectively.

The structure of the remainder of the paper is as follows: Section 2 summarizes the related work. Section 3 introduces the method we used to do the project. Section 4 discusses the results of the project. Section 5 presents the discussion. Section 6 concludes of the paper.

2. Related work

2.1 Web application development methods

Web application development is a field in software development that focuses on building applications that are accessed through web browsers over the Internet. These applications run on web servers and can be used by multiple users simultaneously. Due to the development of technology and increasing demand of online servers, web application development involved greatly over the years. The first version of web applications were simple static web pages with basic HTML in the early 1900s (Jazayeri, 2007). Dynamic content in web application emerged in later 1990s with the appearance of tools like PHP, ASP and JSP, which allow developer to generate more interactive web contents (Jazayeri, 2007). During the mid-2000s, the user can operate and generate contents, and collaborate with other people with the technology of AJAX (Asynchronous JavaScript and XML) (Jazayeri, 2007). During 2000s to 2010s, many web application frameworks and libraries were developed, including Django and jQuery, which they simplified the development process (Jazayeri, 2007). Now, web applications focus more on mobile and responsive web design with the adaptation to various screen sizes and devices (Jazayeri, 2007).

The web application development process consists of a series of steps: design, implement, and deploy, to meet the demand of users and requirement. The first step is to set up the requirement for the web application and define the features of it. The second step is to design the application, including the minimum viable product, the technology stacks, programing languages, frameworks, and so forth we will use for developing the web app. Then, we will create wireframes to visualize the user interface (UI) and user experience (UX), and design the layout of the web pages, color scheme. Next, we will do front-end and back-end development and set up a database to store the data. After the implementation, we will test the web application and deploy it on a web server or cloud platform. Web application development becomes a crucial part of the modern life, in which organization can deliver engaging and interactive experiences to users. Many new techniques and methods will be developed and make the process more efficient and easier to use.

2.1.1 Minimum Viable Product design and development

A minimum viable product (MVP) is a simple version of a diagram. In this development strategy, a product with minimum set of features is developed to satisfy early users and get feedback for further development of the whole product. The goal of the MVP is to validate the product idea with least effort.

The first step of design MVP is identifying the main idea of the project and the problem we will solve. Then, thinking about the core features we will present in this product, and create the process of doing that: input format, output format and data retrieval. After designing it, we can do implementation of it and test the viability of the MVP. After that, we will improve it with feedback from users, to expand its functionality and reduce its limitations one by one.

2.2 Biblioshiny

Biblioshiny is a web application that help researchers perform relevant bibliometric and visual analyses on an interactive web interface, even without coding experience (Aria, 2017). Biblioshiny < https://www.bibliometrix.org/home/index.php/layout/biblioshiny > is developed using the Shiny framework in R, and it provides researchers with easy access to bibliometrix and make the process of network analyze scientific publications simpler. By various visualizations, researchers can analyze citation patterns and conclude important results to the publications. The design and implementation of Biblioshiny is to simplify the process of exploring bibliographic data and help researchers gain valuable insights from the data analysis. The design and development of Biblioshiny is an influential reference for "WhoFundedIt." By investigating the implementation of Biblioshiny, the "WhoFundedIt" web application can also utilize the Shiny framework to display the data and learn from its user interface design and data visualization techniques.

2.3 Application Programming Interface (API)

An API is a set of rules and protocols for building and integrating application software (https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces). An API defines the methods and data formats the application can use to exchange information. APIs are crucial in software development, because they helps developers access the functions or data of existing applications, which makes the developing process easier and efficient.

2.4 Crossref

"Crossref is a nonprofit open digital infrastructure organization for the global scholarly research community. (Rosa-Clark)" Crossref provides Digital Object Identifier (DOIs) and comprehensive metadata for scholarly content, including scientific publications. The Crossref API allows access to plenty of publication data, including author, publications, and funding information. Crossref's data repository is useful for retrieving funder information for a list of

scientific publications in “WhoFundedIt.” The Crossref API ensures that “WhoFundedIt” can offer accurate and updated funder information, making the data retrieval process reliable.

3. Methods

This section describes the approach used to design and develop the “WhoFundedIt” web application to retrieve the funder information for a list of scientific publications. We provide an overview of the Minimum Viable Product (MVP), the data sources, the four technology stacks, the use of APIs, and the implementation strategies of different technology stacks.

3.1 Design the Criteria for the Minimum Viable Product (MVP)

The primary demand of the project was to get funder information to a list of scientific publication. The basic logic of the MVP is that it will input the information of publications, and retrieve funder data from different data sources, and then output the funder information for each publication. Several key factors were taken into consideration when I design the MVP: how to make the product both simplest and viable at the same time. To make the MVP as simple as possible, I decided the input is only the DOI of publications because other bibliography is hard to identify publication. The data only retrieve from Crossref API, because now this is the only data source we can use. The output of the MVP is the funder information in text file. From the viability perspective, I investigated the number of publication have DOI and the number of publication Crossref contains to see whether it is viable.

The input is determined by the most accurate way we can retrieve publication to make sure the viability of it. I evaluate the accuracy of using DOI to retrieve publication is relatively high because DOI is special for each publication and it doesn’t have confusion when we use it. Using other bibliographic information to identify the publication may have some inaccuracy and confusion. For instance, if we use author name, and some of the data source uses the author’s middle name and some don’t, so this may cause some confusion. The input file format is also determined by the simplest way for computer to read file. Text file is one of them that easy to code. The determination of output is according to the demand of users: what do they want to know about the funders. In this simplest version, I think funder name is the most important one to contain. Also, the file format is also as easy as possible, like text file and downloadable.

3.2 Analyze Possible Data Sources

I analyzed several possible data sources for the web application: OpenAlex, Open Citation, Open Payment, Dollors for Doc, and Disclosure UK database. I find them by searching online and see they may contain the funder information for publications. I read through their documentation and APIs to check their availability for “WhoFundedIt.” First, I checked it they can provide funder information for corresponding publication. Then, I tried to call them and see whether we can retrieve data from them.

3.3 Brainstorm and Evaluate Possible Technology Stacks

The first technology stack is the traditional way of web application development, so I just follow the general way to do it. The tool I selected is the ones I used before or they are easy to use. The second one is to design a Shiny app, which is inspired by Biblioshiny. I explored the Shiny framework, and it contains many built-in functions and data visualization samples we can use for this project. The third way, developed as a part of Biblioshiny and add additional functions to Biblioshiny to only take the funder information and show it. This is also inspired by Biblioshiny because it has useful tool to show the network relationship of publications with their metadata. The metadata also includes funder information, so we can just retrieve the funder information and show it. The last one is using neo4j as the output format tool to show the funder information. It will be useful in the future when we improve the web application and specifically show the network analysis of the funder information. The criteria I use to evaluate is first it can be easy to develop, and it is accessible and achievable. It must be simple enough to build the MVP quickly.

3.4 Implement the Chosen Stack

During the implementation process, I developed a small program based on the MVP that can retrieve funder data from Crossref API for a list of DOIs of publication in a text file and then output them in a text file. Then, I explored the Shiny framework and look through the example and samples it provided for the functions. I checked the functions I need for development, which are uploading file, displaying text, downloading file, counting number of funders, plotting the counts of each funder and its name. I added all the functions I need and connect them to the first small program of retrieving funder data.

4. Results: choose one to implement, why; how code address method, design future work

4.1 Criteria for the Minimum Viable Product (MVP)

4.1.1 Minimum Input

The input of the program is only DOI search and all the publications that use the program need to have valid DOI for MVP. The reason I did this is because DOI is a more accurate way to identify and locate the publication. If the scientific publications provide DOI information, then we can just try DOI search first to retrieve the funder data. Also, multiple publication data can be retrieved at the same time currently using a single .txt input file.

4.1.2 Data Source

The data retrieval uses the APIs of different data sources to call the database. My design for the MVP is to retrieve the data from Crossref. Crossref is my first choice because previous work related to this project indicated that Crossref could be a comparably good tool to retrieve funder data.

4.1.3 Minimum Output

After retrieving the funder data, the program will check the accuracy of the data retrieved from Crossref compared to the information given by users. Then, the program will display a log file to show the data retrieval successful rate of the publication. The successful rate will be calculated by the formula: successful rate (SR)= the number of the publication whose funder data have been successfully retrieved / total number of publications that user's input. Then, this data will be displayed to the user. The output file is in .txt format and contains the funder information: id for paper, id for institution, name of institute.

4.2 Possible Data Sources

The primary data source for "WhoFundedIt" was the Crossref API, which provides comprehensive metadata for scientific publications. Crossref includes the Funder Registry, which is a global registry of research funding organizations that provides unique identifiers for funders. By using the DOI of publications from the input file, the program will call Crossref API to retrieve the relevant funder information of each publication.

I investigate other possible data sources, but none of them could retrieve data successfully for different reasons. OpenAlex is a free and open catalog of the world's scholarly papers, researchers, journals, and institutions — along with all the ways they are connected to one another. For the data in OpenAlex, it is difficult to use to retrieve funder data directly and it needs the help of Crossref, because OpenAlex doesn't provide funder data for publications. Also, OpenAlex doesn't have an official API to use to retrieve funder data. According to the official website of OpenAlex, "OpenAlex indexes about 32,000 funders. Funder data comes from Crossref, and is enhanced with data from Wikidata and ROR." So, even if we could use it to retrieve funder data, they all come from Crossref, which we already done. Besides, While the OpenAlex Funder API provide access to funder information, the information is the access to funder metadata instead of the specific funder data associated with individual publications. So currently, we don't have to investigate this data source because it can't solve our problem – how to figure out the funders of publications that are not provided by Crossref API. But we may need this data source in the future when we need to improve the web application.

The other possible data source is Open Payment, which provides transparency about financial relationships between healthcare providers and pharmaceutical companies. It mainly focuses on the payments related to healthcare industry, so it doesn't provide funder information for publications. Also, its dataset is hard to use because everything exist in string and it requires the disambiguation. Similarly, Dollars for Docs has the same problem as Open Payment. Another one is called OpenCitation, which is scholarly infrastructure organization that provides open access to bibliographic and citation data. We can use OpenCitations Index to access citation data and associate metadata. It includes four main datasets, including COC I(citation from Crossref), DOCI(citation from DataCite), POCI(citation from PubMed), and CROCI(citation

from Crowdsourcing). None of these databases contains funder information for publications, so it cannot satisfy our requirements.

4.3 Possible Technology Stacks

The "WhoFundedIt" web application design includes 4 different technology stacks: Web framework and SQL, a Shiny app, as a part of Biblioshiny, neo4j. The technology stacks are carefully chosen to facilitate efficient development for the next step.

The first stack is a traditional method to do web application development, with web framework and SQL databases. The key of this method is creating an individual database for funder information with funder id, name and institution. The data in the database comes from the existing funder database by doing data dump or database merge. Users can access to the database by the API of database. The programming language we plan to use is Python and the backend web framework is Django. Django is a wide-used web framework for Python follows the model, template, views architectural pattern. For the database to store and manage the data, we will use mySQL, one of the most commonly used SQL databases. MySQL is a relational database management system that stores data in separate tables, which fits for the situation of our funder database. Also, the structure of the databased is organized into physical files.

The individual database can be a great resource of funder information, which collects the data from most different data sources. So, if we make one successfully, it will be a great contribution to the research. However, the difficulty of creating an individual database is large. While this approach allows more control over the entire development process, it will be result in a complex development cycle. The number and size of funder data from different databases is large. It is hard to say the data dump will be successful or not. Also, it is hard to update to data. If the funder data is added or changed, the data in this funder database should also be updated. If we do so, we have to think of a way to automatically update the data. Besides, the database can't include all the funder information for each publication in the world. The retrieval successful rate may be low.

The second technology stack is develop a Shiny App by using Shiny framework. Shiny is an open source R package that provides an elegant and powerful web framework for building web applications using R or Python. We will use Shiny environment and R packages to develop the "WhoFundedIt" web application. Also, we will use API queries to call Crossref, one of the funder databases, to retrieve the data. Once the data is retrieved, we will process the data, including cleaning and transforming it. In addition, we will design the user interface of the app that includes elements allowing users to interact with and display the funder data.

Shiny environment is really a useful way to develop the web app. It has a lot of built-in methods and functions for us to use, so it makes the process a lot easier. If we just access the Crossref API instead of store everything with a database, we don't have to worry about the size of data and also the issues about updating. Many R-packages are really useful for implementation. However, Shiny apps have limitation for applications. If there are many users at the same time, it is hard to finish the data processing. The method of retrieving API may cause

more inaccuracy or missing of the result. Also, only Crossref can be used to do this, but the data in Crossref is also limited. If a user requires some publication that is not in Crossref, then the app can't find them. The API accessing to other databases need further exploration.

The third technology stacks is integrating "WhoFundedIt" into the Bilbioshiny web application. We consider write an additional function for Biblioshiny to show the funder information. In Biblioshiny, it analyzes and visualizes the publication relationship and networks with one another. So, the data it has already processed contains the funder information, including funder id and name. We can take advantage of this and write an additional function for Biblioshiny just display the funder information for each publication.

This method makes the process of web app development much easier, because we just need to write an additional function to show the funder information. Biblioshiny is already a successful web app to generate the relationships among the publications and retrieve all the metadata including the funder information. We can look at the code of Biblioshiny as sample, which makes the implementation easier. However, Biblioshiny is the project for their design team. It is hard for us to change something for it. Even though we can write the additional function for it, the function of Biblioshiny is a little bit different than "WhoFundedIt". The feature of Biblioshiny is that the input file from Scopus or Web of Science have already contained the funder information, and the function of it is just to display the information. If we want to just visualize this part, and it works. But I think the main goal for our project is to get the information from different sources, instead of with the input file in just one data source. So, the idea of adding an additional function to Biblioshiny may not work well and have some limitations.

The fourth stack is using Neo4j to display the output in network analysis. This method is to use graph database GraphDB, to do the network visualization of the publication and funder. We can store and query data structurally, like the relationship between funders, publications, authors. I use Shiny environment for this as well to make it easier and maybe we can incorporate this with Stack 2 together. It can generate the network visualization and give more function to users. But if we want only to display the name instead of the network relationship, it is will not be that useful. So, this approach may work in the future when we need more functions for "WhoFundedIt," but currently, It is not the core functionality we pursue.

4.4 Implementation of the Chosen Technology Stack

After careful consideration, the optimal approach is to build an independent Shiny app for "WhoFundedIt" and I put it as the first one to do implementation. That's because Shiny is a framework that easy build interactive web apps from Python. We don't have to have HTML, CSS, or JavaScript skills, so it can simplify the development process. Also, it is extendable and flexible with many integrations. The functionality in Shiny is useful in out app because it can easily do data visualization and network analysis with many sample code.

The MVP of the "WhoFundedIt" web application is the simplest version with very basic functionality of the product. The main task of it is to retrieve the funder data for the publications

with DOI from Crossref API. The MVP of this project can only be a program instead of a web application if we won't store funder information on cloud storage. The program still needs an Internet connection to run.

5. Discussion:

5.1 Limitation for Work

The input of the program requires every publication to have DOI (Digital Object Identifier). If the publication doesn't have DOI or its DOI can't retrieve anything, the program can't get the funder information for them. Also, the input of the program only accepts a .txt file. Besides, this version of the program just uses Crossref as the data source because it has DOI and funder registry. The output file is only .txt file.

5.2 Future Work

For the future work, we have to solve the limitations of the design for MVP independently. If the publications don't have a correct or complete DOI, we will add another function for the input publication without a DOI or the case that DOI can't retrieve anything. The input value will be the bibliography of the publication with as complete information as possible with author, article title, publication date, journal title, volume/issue, page range. Let's discuss the simplest way to input the bibliography. Title is not really a good option to locate the publication because the format and different ways of writing it may cause issues. So, we can use fuzzy match for title, but it cannot be used in our case because this is only the minimal viable product. Publication year can be an option to locate the publication, but not a very optimal one, because the range is too big and it is hard to locate the right one. Author names also can be a way to locate, but the issue it may cause is some non-English names and also the confusion of middle names. The others, journal name, publication data, page number, ISSUE number are hard to be wrong, so the simplest and better way to input the metadata is the combination of journal name, author names, publication data, page number and ISSUE Volumes. Also, the input file should also be various: .txt, BibTeX, .csv. Besides, if Crossref doesn't have the record of the publication, we will search for other data sources: OpenAlex, Open Payment, Dollar for Profs, Funder Registry, OpenCitation, Disclosure UK database (I put the data sources in the order I believe we can use according to its usefulness). Now, most of those data sources need to be explored and investigated. So, this part needs to be updated when we have more information about the data sources. Data Integration of those data sources should also be added to improve the program. Currently, only Crossref will be used as the database, but later if we add more database, we should do a better data integration to combine the data from different databases.

Output file can be in more formats. Currently, we only consider the simplest version, .txt file. We will add more output file format: .pdf, .csv, network visualization.

6. Conclusion:

In conclusion, the “WhoFundedIt” web application is a significant and valuable resource for researchers in academics. It can help researchers identify potential funders for their project and promote the integrity in research. By considering technology, data sources, and user-friendly design, “WhoFundedIt” support researchers to make a impact in their field and make the research easier.

7. Code availability:

https://github.com/infoqualitylab/WhoFundedIt-app

8. Acknowledgements

The work was supported in part by the Distributed Research Experiences for Undergraduates (DREU) program, a joint project of the CRA Committee on the Status of Women in Computing Research (CRA-W) and the Coalition to Diversify Computing (CDC), which is funded in part by the NSF Broadening Participation in Computing program (NSF BPC-A #1246649). Mentors were supported by NSF 2046454, CAREER: Using network analysis to assess confidence in research synthesis.

References

Aria M, Cuccurullo C (2017). “bibliometrix: An R-tool for comprehensive science mapping analysis.” Journal of Informetrics. https://doi.org/10.1016/j.joi.2017.08.007.

Krimsky, S. (2013). Do financial conflicts of inteterest bias research? an inquiry into the "funding effect" hypothesis. Science, Technology, & Human Values, 38(4), 566–587.

Kee, M., Greenough, M., Anderson, J. M., Weaver, M., Hartwell, M., & Vassar, M. (2021). Authorial conflicts of interest and sponsorship in systematic reviews and meta-analyses on psoriasis. Journal of Psoriasis and Psoriatic Arthritis, 6(4), 174–184. https://doi.org/10.1177/24755303211020677

Jazayeri, M. (2007, May). Some trends in web application development. In Future of Software Engineering (FOSE'07) (pp. 199-213). IEEE.

Rosa-Clark. (n.d.). About us - Crossref. www.crossref.org. https://www.crossref.org/community/about/