

CMPUT 499: Mining Software Repositories

Literature Review

Monica Bui

# 1 Introduction

Third-party software libraries and Application Programming Interfaces (APIs) offer a way for developers to use existing features and functionalities to build their projects without having to re-invent the wheel. There are many libraries to use for different programming languages whether it is open source or hosted elsewhere. Although having this much freedom for deciding on what libraries to use is flexible, it is also conflicting to figure out which one is best for your project. It is difficult to choose because there are a variety of factors involved such as functionality and developer support. In this review, we look at several different articles to help analyze various properties of libraries and to determine a helpful comparison between them.

## 2 Article Review

### 2.1 Library Recommenders

One problem is researching new analogical libraries to use for different programming languages that has similar functionalities to the ones you currently know. Chen's et. al [1] paper discusses their library recommender that compiles a list of libraries from community resources such as blogs and Q&A sites like StackOverflow [2] and outputs a list of recommended libraries in the developer's language of choice. This is implemented by first mining tags on questions posted online on StackOverflow [2]. These tags are split into two knowledge bases: relational and categorical. Respectively, relational knowledge is how pairs of tags are correlated to each other eg. Java and JUnit while categorical knowledge consists of how tags are grouped into categories such as language, operating system, concept, or library with both bases being analyzed by NLP. The key idea here is that having different separated tag categories and relationships between tags often mentioned together allowed for a simpler way to recommend a new library. With the database in place, users can then search for recommendations through the built web application called SimilarTech [3]. While trying out the application myself, I found that search results would only yield for libraries that are mentioned in specific contexts. Axios [4] a popular Javascript HTTP library should output expected recommendations like Requests [5] module for Python but the actual list printed was empty. While the number of languages it can suggest li-

braries for is limited to 5, the precision metric is impressive with 1 language at 81% and with 5 being at 67% showing its potential to grow in the future.

Going back to the key problem of choosing the right library to use, Uddin's et. al [6] article highlights their approach to this by looking at personal developer opinions's on different resources and how it affects the reader's decision. Furthermore, the sentiment behind this can be used to indicate if its a positive, questionable, or a negative API to use. The tool created, Opiner [7] is a summarization engine that examines these opinions and evaluates its sentiment to see if the API should be recommended also displaying ratings on API traits and organizing top opinions on both positive and negative sides. The backend of the application web crawls through a Q&A site, StackOverflow [2] to extract answer information surrounding an API topic. This data is used to help discover when an API is mentioned, what kinds of opinion are associated with an API, and combining these opinions under common aspects developers care about such as usability and performance. Using Opiner [7] through a small research study evaluated on 2 separate groups of participant's choices to see if just StackOverflow [2] alone would work for decision making on selecting an API or both StackOverflow and Opiner would be better. We view that developers are more confident in their selection with both tools vs just StackOverflow alone. A weakness is that although the evaluation of the study is optimistic, having 100% as a metric for using both would be hard to generalize for a larger population. Giving a hard edge case percentage will be difficult to tell later on for false positive tests.

## **2.2 Github Badges**

TODO) Talk about NPM repo badge article

## **2.3 Metrics**

TODO) Talk about Fernando's article

### 3 Conclusion

## References

- [1] C. Chen, S. Gao, and Z. Xing, “Mining analogical libraries in q & a discussions – incorporating relational and categorical knowledge into word embedding,” in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, vol. 1, March 2016, pp. 338–348.
- [2] Stack-Overflow, “Stack overflow,” <http://www.stackoverflow.com>.
- [3] Similartech, “Similartech,” <https://graphofknowledge.appspot.com/similartech>.
- [4] Axios, “Axios,” <https://github.com/axios/axios>.
- [5] Requests, “Requests,” <https://github.com/requests/requests>.
- [6] G. Uddin and F. Khomh, “Opiner: An opinion search and summarization engine for apis,” in *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Oct 2017, pp. 978–983.
- [7] Opiner, “Opiner,” <http://sentimin.soccerlab.polymtl.ca:38080/opinereval/>.