Project Proposal:
Study the knowledge flow to Stack Overflow and from Stack Overflow to GitHub

Wenhan Zhu (Cosmos)
w65zhu@uwaterloo.ca
University of Waterloo
Waterloo, Canada

ABSTRACT

Stack Overflow is the most popular Q&A website that covers content about programming. It is considered to be consisted of large amount of public knowledge. Many answers from StackOverflow has been used in projects on *GitHub* and many of the answers originates from other places. It is important that we understand how knowledge transfer into *Stack Overflow* and how other places such as *GitHub* uses information from *Stack Overflow*.

KEYWORDS

Stack Overflow, *GitHub*

1 INTRODUCTION

Stack Overflow since it's introduction in 2011 has became the largest Q&A website that focuses on programming. It has been considered to have a large amount of public knowledge and extracting information and understanding *Stack Overflow* has been quite popular in the Software Engineering community. There has been numerous studies on *Stack Overflow*, people try to understand how it works and predict quality answers. Recently, there has been a trend to extract *Stack Overflow* knowledge to help developers. [7] Recently, a new database *SOTorrent* based on the official *StackOverflow* datadump has been created.[2]

GitHub is the largest code sharing website that hosts *git* repositories. There are many code segments on *GitHub* that references from *Stack Overflow*. Previous studies has shown that code reuse on *GitHub* from *Stack Overflow* could cause security issues. [6]

The goal of this study is to learn more about how knowledge gets shared to *Stack Overflow* and how it then gets used on *GitHub* projects.

2 MOTIVATION

Stack Overflow is very popular among developers, study has shown that code reuse from *Stack Overflow* is very common. However, not much has been studied on this topic. The flow of the information and its evolution throughout the process could help us understand better how public knowledge is shared and evolved during software development.

Stack Overflow dumps its data into a database in schedule. The database contain meta information about posts. *SOTorrent* is a recent database that's based of the official data dump. It adds more information about links referenced on *Stack Overflow* and *GitHub* references to *Stack Overflow*. With this information we could trace the information flow on *StackOverflow*.

3 DATASET

There are 3 major resources that would be used by this project. The first is the before mentioned *SOTorrent*. *SOTorrent* is a database build on top of the official *Stack Overflow* data dump. The additional tables in *SOTorrent* is a classified code/text block relationship between updates of posts. And information about references in *Stack Overflow* and *GitHub* references to *Stack Overflow*.

The database is composed of *CSV, XML* files that contain information about *Stack Overflow* posts content and meta data. *Stack Overflow* as an Q&A website has questions and answers, they are all considered posts. They are differentiated in the dataset by their post type. Each post can be upvoted or downvoted by users. The voting information also exists in the dataset.

The other major dataset will be the *GitHub* dataset. There are two main choices, *GHTorrent* and the *Google github_repos* dataset. At the moment, there needs to be more investigation into the dataset to determine which one is more suitable for the purpose of this study.

The other resource will be a code clone detection tool that supports detecting the changes during the clone. Since the project will evaluate how information from *Stack Overflow* is used on *GitHub*, it is desired to have a way to identify the usage and check the differences if there are any.

4 RESEARCH QUESTIONS

I would like to investigate in the following research questions:

(1) RQ1: What does *Stack Overflow* information come from?
(2) RQ2: How are *Stack Overflow* code used on *GitHub*?
(3) RQ3: Are *Stack Overflow* code modified when introducing to *GitHub*?
(4) RQ4: Does *Stack Overflow* code evolve after introduction to *GitHub*?
(5) RQ5: How does *Stack Overflow* code on *GitHub* co-evolve together?

4.1 Approaches

RQ1: What does *Stack Overflow* information come from?

To tackle this problem, my current plan is to extract the first and second level domains from the reference dataset and determine the type of the website. The answer would ideally be some sources from websites and its types such as forums, papers or course notes.

RQ2: How are *Stack Overflow* code used on *GitHub*?

The usage of *Stack Overflow* code on *GitHub* has been studies before. *Baltes et al.* looked at how the top 10 most reference *Java* posts on *Stack Overflow* are used on *GitHub*. [1] However, none of them looked at the depth of the question. I would like to look at the question at another direction. The goal is to find out what kind of repos uses *Stack Overflow* code, are they all course projects by students or its equally distributed among all repositories.

RQ3: Are *Stack Overflow* code modified when introducing to *GitHub*?

It is often needed to modify the code when reusing code. Previous study suggests that code snippets on *Stack Overflow* is not always reusable. [11] The goal of this question is to have a better understanding of code reuse on *GitHub* from *Stack Overflow*. The approach is to have a code clone tool to look at the 2 code segments and determine are there any changes. After this, we could have an overview of how many code are directly copied and how many are modified.

RQ4: Does *Stack Overflow* code evolve after introduction to *GitHub*?

Code evolves over time. By looking at this problem, we can see how code evolves after reused from *Stack Overflow* and provides information for the next research question. I would like to approach this question by looking at commits updates regarding the lines of code reused from *Stack Overflow*.

RQ5: How does *Stack Overflow* code reused on *GitHub* co-evolve together.

Previous studies have shown that code snippets on *Stack Overflow* are not very reliable in terms of security issues. [6] When developers updates the insecure code segments, do they go back to the original *Stack Overflow* posts provide the information? These are questions that I would like to explore.

5 RELATED RESEARCH

Since *Stack Overflow*'s introduction, it has attracted many researchers to work on the subject. Some research wants to understand the model of *Stack Overflow* [3] and why it works with its gamification. [4] Interaction between *Stack Overflow* and other platform has also been looked at. *Vasilescu et al.* looked into common users between *Stack Overflow* and *GitHub*. They discovered that novice and experienced developer from *GitHub* have different patterns of activitiy on *Stack Overflow*. [10]

There has also been research done to study how to improve *Stack Overflow*. *Duijn et al.* looked into prediction the quality of questions by looking at how many answers the question attracted. [5] Previous study has shown that the number of answers of a question is highly correlated to the number of upvotes. Therefore, *Dujin et al.* decided to use 2 answers (the mean of answers to *Stack Overflow* posts) as a line for quality questions. *Saha et al.* created a model to better predict tags for *Stack Overflow* questions. [8]

Researchers also use *Stack Overflow* data to improve the development process of developers. *Ponzanelli et al.* tried to mine *Stack Overflow* for better IDE suggestions. [7]

Mining *Stack Overflow* has been really popular in recent years. The public knowledge from *Stack Overflow* is very rich. *Treude et al.* mined information from *Stack Overflow* to augment API documentation. [9]

6 VULNERABILITIES

6.1 Threats to Validity

Both the reference to and from *Stack Overflow* in *SOTorrent* are extracted by looking at *HTML* links. So there could be referenced information but not present in the dataset. Also not every code reuse is referenced, for code reuse from *Stack Overflow* to *GitHub* we could look at site wise code clone detection but the scale will be enormous and unlikely to be possible due to the size of the corpus. There won't be any effective way off my mind how we would detect unreferenced sources to *Stack Overflow*.

The source clone tool that is landed on could lead to problems. So I'll try my best to find the tool that serves the purpose will.

6.2 Potential Problems

In this case, the source code is required from *GitHub*. Due to the API limits of *GitHub* it's unsure whether acquiring the source code will be smooth.

As a novice to databased and previous problems I've encountered with them. Having the dataset setup could also be more effort than expected.

Having not used a source clone tool before, experimenting and setting up could have problems.

Time could also be a problem due to the scale of the project. However, I would like to answer most of the questions.

REFERENCES

[1] Sebastian Baltes and Stephan Diehl. 2018. Usage and attribution of Stack Overflow code snippets in GitHub projects. *Empirical Software Engineering* (01 Oct 2018). https://doi.org/10.1007/s10664-018-9650-5

[2] Sebastian Baltes, Lorik Dumani, Christoph Treude, and Stephan Diehl. 2018. SOTorrent: Reconstructing and Analyzing the Evolution of Stack Overflow Posts. In *Proceedings of the 15th International Conference on Mining Software Repositories (MSR '18)*. ACM, New York, NY, USA, 319–330. https://doi.org/10.1145/3196398.3196430

[3] Anton Barua, Stephen W. Thomas, and Ahmed E. Hassan. 2014. What are developers talking about? An analysis of topics and trends in Stack Overflow. *Empirical Software Engineering* 19, 3 (01 Jun 2014), 619–654. https://doi.org/10.1007/s10664-012-9231-y

[4] Huseyin Cavusoglu, Zhuolun Li, and Ke-Wei Huang. 2015. Can Gamification Motivate Voluntary Contributions?: The Case of StackOverflow Q&#38;A Community. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work &#38; Social Computing (CSCW'15 Companion)*. ACM, New York, NY, USA, 171–174. https://doi.org/10.1145/2685553.2698999

[5] Maarten Duijn, Adam Kučera, and Alberto Bacchelli. 2015. Quality Questions Need Quality Code: Classifying Code Fragments on Stack Overflow. In *Proceedings of the 12th Working Conference on Mining Software Repositories (MSR '15)*. IEEE Press, Piscataway, NJ, USA, 410–413. http://dl.acm.org/citation.cfm?id=2820518.2820574

[6] F. Fischer, K. Böttinger, H. Xiao, C. Stransky, Y. Acar, M. Backes, and S. Fahl. 2017. Stack Overflow Considered Harmful? The Impact of Copy amp;Paste on Android Application Security. In *2017 IEEE Symposium on Security and Privacy (SP)*. 121–136. https://doi.org/10.1109/SP.2017.31

[7] Luca Ponzanelli, Gabriele Bavota, Massimiliano Di Penta, Rocco Oliveto, and Michele Lanza. 2014. Mining StackOverflow to Turn the IDE into a Self-confident Programming Prompter. In *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014)*. ACM, New York, NY, USA, 102–111. https://doi.org/10.1145/2597073.2597077

[8] Avigit K. Saha, Ripon K. Saha, and Kevin A. Schneider. 2013. A Discriminative Model Approach for Suggesting Tags Automatically for Stack Overflow Questions. In *Proceedings of the 10th Working Conference on Mining Software Repositories (MSR '13)*. IEEE Press, Piscataway, NJ, USA, 73–76. http://dl.acm.org/citation.cfm?id=2487085.2487103

[9] C. Treude and M. P. Robillard. 2016. Augmenting API Documentation with Insights from Stack Overflow. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. 392–403. https://doi.org/10.1145/2884781.2884800

[10] B. Vasilescu, V. Filkov, and A. Serebrenik. 2013. StackOverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge. In

2013 International Conference on Social Computing. 188–195. https://doi.org/10.1109/SocialCom.2013.35

[11] Di Yang, Aftab Hussain, and Cristina Videira Lopes. 2016. From Query to Usable Code: An Analysis of Stack Overflow Code Snippets. In *Proceedings of the 13th International Conference on Mining Software Repositories (MSR '16)*. ACM, New York, NY, USA, 391–402. https://doi.org/10.1145/2901739.2901767