
A Gaze-Based Exploratory Study on the Information Seeking Behavior of Developers on Stack Overflow

Cole S. Peterson

University of Nebraska - Lincoln
Lincoln, Nebraska, USA
cole.scott.peterson@huskers.unl.edu

Jonathan A. Saddler

University of Nebraska - Lincoln
Lincoln, Nebraska, USA
jsaddle@cse.unl.edu

Natalie M. Halavick

Purdue University
West Lafayette, Indiana, USA
nhalavic@purdue.edu

Bonita Sharif

University of Nebraska - Lincoln
Lincoln, Nebraska, USA
bsharif@unl.edu

ABSTRACT

Software developers use Stack Overflow on a daily basis to search for solutions to problems they encounter during bug fixing and feature enhancement. In prior work, studies have been done on mining Stack Overflow data such as for predicting unanswered questions or how and why people post. However, no work exists on how developers actually use, or more importantly, read the information presented to them on Stack Overflow. To better understand this behavior, we conduct an eye tracking study on how developers seek for information on Stack Overflow while tasked with creating human-readable summaries of methods and classes in large Java projects. Eye gaze data is collected on both the source code elements and Stack Overflow document elements at a fine token-level granularity

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland UK

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5971-9/19/05.

<https://doi.org/10.1145/3290607.3312801>

KEYWORDS

eye tracking study; Stack Overflow; source code; reading behavior; code summarization

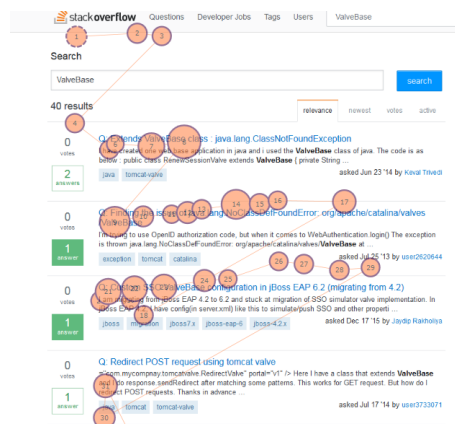


Figure 1: The gaze plot of a developer viewing the results of a search on Stack Overflow. The circles represent fixations. A fixation is a stabilization of the eye gaze on a certain area for a period of time - typically 100 milliseconds.

using iTrace, our eye tracking infrastructure. We found that developers look at the text more often than the title in posts. Code snippets were the second most looked at element. Tags and votes are rarely looked at. When switching between Stack Overflow and the Eclipse Integrated Development Environment (IDE), developers often looked at method signatures and then switched to code and text elements on Stack Overflow. Such heuristics provide insight to automated code summarization tools as they decide what to give more weight to while generating summaries.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; **Empirical studies in collaborative and social computing**; *Collaborative and social computing*; • **Software and its engineering** → *General programming languages*;

INTRODUCTION

Stack Overflow (SO) is a widely used question and answer website for developers. Software developers use it to post questions and answers related to programming and software engineering problems. Questions commonly asked include how specific Application Programming Interface (API) elements are used or efficient methods in coding a particular program. Stack Overflow is a widely researched dataset, but there is not much research on how developers use this resource. To date, it includes over 12 million questions, 19 million answers and 47 million comments.

In this paper, we present a pilot exploratory study on how developers seek information on Stack Overflow. The task given was to summarize two classes and two methods in Java open source systems. We used an eye tracker to determine which elements developers were looking at in both Stack Overflow and source code during the study. We chose the task of code summarization because documentation is not always up to date and developers rarely change documentation when changing functionality. Our overarching goal is to learn via eye tracking how developers go about summarizing code using Stack Overflow. We can use those strategies to create heuristics that feed into an automated code summarization tool that creates readable summaries using different contexts such as source code and SO posts. The study presented here is a first step in that direction.

To utilize Stack Overflow to its full capacity, a developer must know not only how to search for relevant questions, but also which parts of the question are most indicative of a good question and answer. To this end, we address the following research questions.

- RQ1: What parts of the Stack Overflow questions and answers do developers focus on most?
- RQ2: What elements do developers transition between on SO posts and the Eclipse IDE?

Table 1: Methods and Classes in the Study. The Task: “Summarize the implementation and usage of the following method/class.”

Element	Description
Method	android.app.Dialog.onSearchRequested()
Class	android.widget.Chronometer
Method	org.eclipse.swt.widgets.Widget.dispose()
Class	org.eclipse.swt.widgets.Dialog

RELATED WORK

Many researchers have used Stack Overflow’s data as a predictive tool such as recognizing the gender of the users [6]. Other researchers have studied how developers ask questions. This includes what types of questions are answered, who answers questions, and how are *good* answers selected [7]. Novielli et al. studied how certain qualities of a question contribute to the success of a question on Stack Overflow [1]. They found a successful question tends to have a code snippet, good presentation quality, and a low quantity of uppercase characters. Ford et al. investigated the involvement of peers on Stack Overflow. She found that gender barriers exist across Stack Overflow, but they are more prevalent in influencing women’s behavior [2]. Women are also more likely to repost if they initially interact with another woman. Some previous studies focused on finding where and what kind of information developers look for and what was preventing them from finding this information [5], however Stack Overflow was not used. As far as we are aware, this is the first study using eye tracking to study a developer’s behavior on Stack Overflow during (not after) the task. We chose eye tracking as a method of data collection as it tells us the thought process of a developer as the task is happening without added effort (unlike think-aloud studies) [4].

STUDY DESIGN

We briefly describe the study tasks, participants, data collection and study instrumentation. In this study, each participant was asked to provide a typed summary of two methods and two classes (See Table 1) - chosen from the Android and Eclipse code repositories - in human-readable English sentences. In particular, they were asked to summarize the implementation and usage of the methods and classes. We chose these methods and classes from a larger initial set because there was enough discussion about them in the Eclipse and Android Stack Overflow pages. See Figure 1 for one developer’s reading of search results on Stack Overflow.

During summarization, the participants were asked to use Stack Overflow as well as the Java source code provided to them within the Eclipse Integrated Development Environment (IDE). This setup mimics real-life development in the field by creating a realistic representation of how developers use Stack Overflow while solving their own problems. We gave only the base URL of StackOverflow.com to participants as a prompt to begin searching. They were able to create their own queries to find information about the two methods and classes. To avoid any confounding factors, we removed any prior existing comments from the entire codebase.

We used the Tobii X-60 eye tracker to collect eye tracking data on both the source code and Stack Overflow elements within the Eclipse IDE and the web browser respectively. We used the eye tracking infrastructure iTrace [3] (www.i-trace.org), that connects to an eye tracker and automatically maps eye gaze on semantically meaningful elements in the code (if statements, identifiers ...) and in Stack

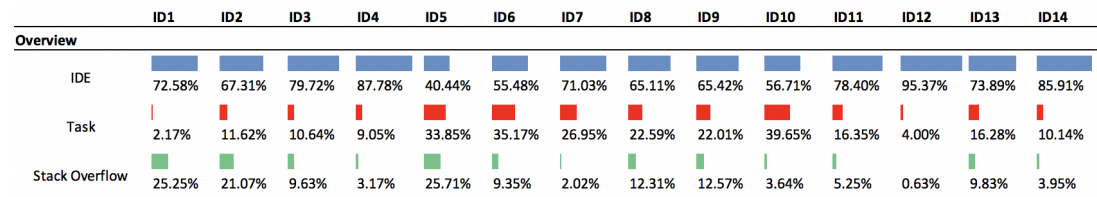


Figure 2: Overview of Gazes per Participant Distributed by Time Spent Looking at Each Context

Overflow (title, description, code, images, comments, etc...). This mapping works in the presence of scrolling and context switching. Currently, iTrace is the only tool that can do this automatically. We used an eye tracker instead of think-aloud studies or questionnaires because it can capture significantly more fine-grained information during the task without interruptions [4]. After a short calibration, the participant was ready to start the tasks. We ran this study with fifteen Computer Science senior students in an eye tracking lab. All participants were familiar with Java and the Stack Overflow website. The study took approximately 30 minutes to complete.

DATA PROCESSING

The gaze data was first processed using a static analysis tool, srcML (www.srcml.org) that maps the gaze and its duration to specific tokens on lines in Java source code. After this initial processing, the data was filtered and aggregated to find the distributions of the time spent looking at Stack Overflow, the Eclipse IDE environment, and the task text file where the participants wrote the summaries. We then calculated the distribution of gaze transitions between element types on Stack Overflow and the IDE. We discarded data from one participant as he did not use Stack Overflow to complete the task due to some difficulty understanding the instructions.

PRELIMINARY RESULTS

The overall distribution of gaze time between the Eclipse IDE, Stack Overflow, and the task file where participants wrote their summaries is shown in Figure 2. We see that the participants spent most of their time looking at the code base in the Eclipse IDE however almost all of them read SO pages for the task. We report gaze time in aggregate on all tasks combined. A task by task analysis is part of our ongoing work. The results of the study find several observations in response to RQ1 seen in Figure 3. On Stack Overflow, *text* and *code* are the elements that each participant spent a majority of their time looking at for both questions and answers. In addition, we don't see a large difference in the distribution of a participant's time between questions and answers. The main difference that can

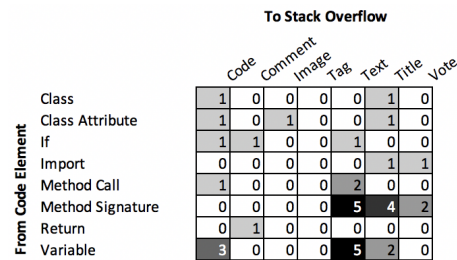


Figure 4: Sum of all participants' transitions from Java elements to the Stack Overflow elements with darker shades representing a more frequently seen transition.

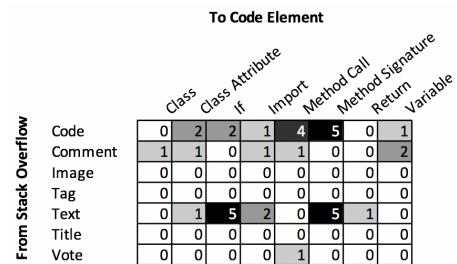


Figure 5: Sum of all participant's transitions from Stack Overflow elements to Java elements with darker shades representing a more frequently seen transition

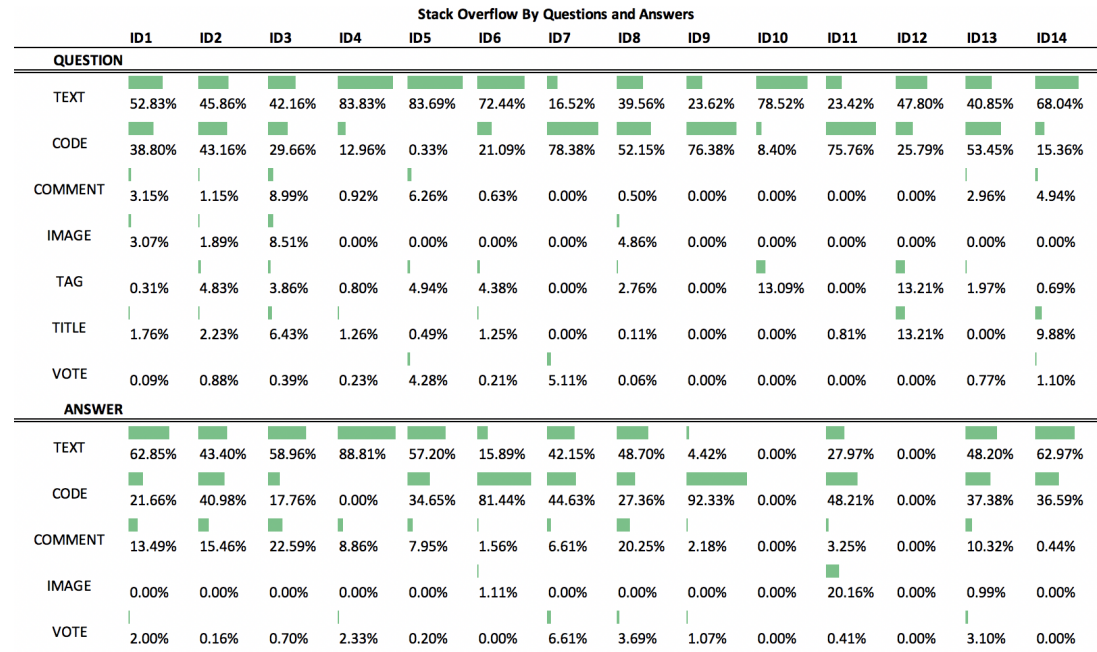


Figure 3: Gaze Duration Distribution per Participant on Stack Overflow Elements

be seen is a larger proportion of time being spent looking at comments in answers. Votes are rarely looked at in both questions and answers in Stack Overflow, with a maximum of 6.61% of the time looking at answers and a maximum of 5.11% of the time looking at the questions (ID7). Overall, we found participants to read the *text* more than the *title* of the post and *code* more than *comments*. They rarely looked at *tags* or *votes*.

To answer RQ2, the gaze transitions between the Java source code and Stack Overflow questions were filtered and gathered into a transition matrix. Transitions from Java source code and the Stack Overflow page elements are shown in Figure 4. The most common transitions from Java to Stack Overflow are *method signatures* to the question/answer *text* or *title* of a post and from a *variable declaration* to the question/answer *text* in a SO post. *Method signatures* are one of the most looked lines of Java source code when returning from SO pages. Besides *method signatures*, when leaving SO, participants mostly looked at either the question/answer *text* element or *code* element before

returning their gaze to the Java source code (see Figure 5). As part of future work, we are looking at *when* these transitions occurred in order to relate that knowledge to when the tasks were completed.

CONCLUSIONS AND FUTURE WORK

The paper presents our initial results on what developers look at on Stack Overflow and how they navigate between source code and SO pages when summarizing code elements. The title of a post was not looked at as much as the rest of the text on the SO post and rarely did participants look at tags or votes.

As part of our ongoing future work, we are analyzing the summaries that the participants wrote and correlating them to the specific content looked at on SO pages. We are also conducting an in-depth analysis - per method and class summarized - of what the participants looked at while they read the code. We also plan to analyze the queries participants used during the SO search and how they relate to task correctness, completeness and efficiency. Such information is very useful for automated code summarization tools as they pick and choose what keywords to include in summaries. The findings on how developers search will not only help increase the quality of questions and answers on Stack Overflow via community guidelines, but also the success rate of finding a relevant resource to developers' problems. In addition, this information could help educators teach students how to look for relevant information on Stack Overflow.

REFERENCES

- [1] Fabio Calefato, Filippo Lanubile, and Nicole Novielli. 2018. How to ask for technical help? Evidence-based guidelines for writing questions on Stack Overflow. *IST Journal* 94 (2018), 186–207. <https://doi.org/10.1016/j.infsof.2017.10.009>
- [2] Denae Ford, Alisse Harkins, and Chris Parnin. 2017. Someone like me: How does peer parity influence participation of women on stack overflow?. In *IEEE Symp. VL/HCC 2017, Raleigh, NC, USA*. 239–243. <https://doi.org/10.1109/VLHCC.2017.8103473>
- [3] Drew T. Guarnera, Corey A. Bryant, Ashwin Mishra, Jonathan I. Maletic, and Bonita Sharif. 2018. iTrace: eye tracking infrastructure for development environments. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, ETRA 2018, Warsaw, Poland, June 14-17, 2018*. 105:1–105:3. <https://doi.org/10.1145/3204493.3208343>
- [4] Katja Kevic, Braden M. Walters, Timothy R. Shaffer, Bonita Sharif, David C. Shepherd, and Thomas Fritz. 2015. Tracing Software Developers' Eyes and Interactions for Change Tasks. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2015)*. 202–213.
- [5] Andrew J. Ko, Robert DeLine, and Gina Venolia. 2007. Information Needs in Collocated Software Development Teams. In *29th Intl Conf on Software Engineering (ICSE 2007), Minneapolis, MN, USA, May 20-26*. 344–353. <https://doi.org/10.1109/ICSE.2007.45>
- [6] Bin Lin and Alexander Serebrenik. 2016. Recognizing gender of stack overflow users. In *Proc. of the 13th Intl Conf on Mining Software Repositories, MSR 2016, Austin, TX, USA, May 14-22, 2016*. 425–429. <https://doi.org/10.1145/2901739.2901777>
- [7] Christoph Treude, Ohad Barzilay, and Margaret-Anne D. Storey. 2011. How do programmers ask and answer questions on the web?. In *Proceedings of the 33rd International Conference on Software Engineering, ICSE 2011, Waikiki, Honolulu, HI, USA, May 21-28, 2011*. 804–807. <https://doi.org/10.1145/1985793.1985907>

OBSERVATIONS

The code snippets are the second most looked at element on SO questions and answers

Even though a participant could freely search SO posts to write their summaries, they still spent a lot of time looking at the code itself. Figure 2 shows the distribution of time spent in the IDE vs. writing the summary (task) vs. SO pages.

Comments on answers to questions are looked at more than comments on the question itself.

Tags and Votes were rarely looked at in this study. It is possible that the task we used (summarization) might have caused this. Another task could have a different result.

ACKNOWLEDGMENTS

We thank all the participants who participated in this study. This work is supported in part by grants from the National Science Foundation under grant numbers CCF 18-55756 and CCF 15-53573.