How Developers Read & Comprehend Stack Overflow Questions for Tag Prediction

Youngstown State University
Senior Capstone project By Ali Morris

I. Abstract

This study is a subset of a larger project being conducted at Youngstown State University. The larger project is looking into how tags can be predicted for Stack Overflow Questions using eye-tracking and machine learning algorithms. This project is a focus on studying gaze behavior of developers when evaluating Stack Overflow Questions. Gaze data such as fixation count, fixation duration, and utilizing AOIs will be used to analyze collected data. The outcomes of the study will focus on where developers focus, valuable areas of interest, and how these studies can be used in the field of auto generating tags.

II. Introduction

a. Objectives & Research Questions

The objectives for this study are:

- Determine what developers focus on when reading Stack Overflow questions to assign tags using eye-tracking
- Determine valuable areas of interest (AOIs) for tag assignment especially keywords

The research questions for this study are:

RQ1. Which sections, such as code, of posting are most valuable when assigning tags?

RQ2. How will non-novice developers compare against novice developers in regards to tag assignment accuracy, reading patterns, and areas of interest?

RQ3. How can this information be used to enhance existing auto-generation tag techniques?

b. Stack Overflow

Stack Overflow is the website that this study is being conducted on. Stack Overflow is the largest online community for programmers to learn and share their knowledge. To put into perspective the size of this website, it consists of 2 million questions, 19 million answers, and 47 million comments. The content is available for download at a data dump size of 70GB.

Stack Overflow is in open-forum format, developers are able to post questions and other developers are able to respond to this postings with solutions. The size and popularity of the website makes it a widely used tool for both student and professional developers alike. The organization of this website is dependent on a tagging system. Tags are words or phrases that relate to the posting. Stack Overflow has a list of legal tags and users with more seniority and experience are allowed to suggest tags not from the list. The accuracy of the tagging system directly affects the usefulness of Stack Overflow as a tool. If postings do not have the appropriate tags it makes it hard to locate the solutions.

c. Related Work

There exists current studies on auto-generation of tags for Stack Overflow questions. All the studies reviewed for this project did not use eye-tracking. The current best tag accuracy is 68.47%[1]. These studies use data mining and machine learning algorithms. Though the studies were not identical, they mostly consisted of data mining many postings from a data-set to extract important features and tags. Machine learning algorithms were trained to recognize these features and corresponding tags. Then when new postings are created their tags are assigned based on previous posts that have similarities [1]-[3]. While these studies are useful, the accuracy can be improved using eye-tracking technologies.

d. Eye-Tracking Field

Using gaze data as implicit feedback can provide very useful information. Gaze data holds information about visual attention including thought processes, strategies, and user technique. Using eye-tracking to study how developers read and work is a relatively new field. Eye-tracking is able to capture a huge amount of data per session: running at 60 Hz an eye tracker can capture up to 60 samples per second. This data can be very insightful to where and why developers are focusing. Different types of gaze data holds different types of information.

Some of the most widely used types of gaze data include:

- Fixations: focus point where the eyes remain stationary for some time. These can be counted.
- Duration: total amount of fixation time for an area
- Saccade: Quick eye movement between fixations
- Scanpath; sequences of saccade-fixation-sacaade that interconnect
- Areas of Interest (AOI): Specific areas on the screen on which quantitative eye movements are calculated

In this study fixation count, fixation duration, and AOIs will be used.

III. Experiment Design

The experiment was conducted at Youngstown State university in the eye-tracking lab. The labs use Tobii Studio for experiment conduction as well as analyzing. There were seven participants total consisting Computer Science, CIS graduate students, and Electrical Engineers all attending YSU. The participant's experience ranged from less than a year and up to five years. Each was briefed on the study and participated in pre-surveys and post-surveys. These studies were helpful in analyzing the data and later grouping the participants.

Each participant reviewed 9 tasks. Tasks consist of a Stack Overflow question sourced directly from the site as well as a "Suggested Tags" list. The suggested tags list is a randomized list consisting of 5 relevant tags and 5 distractor tags. Relevant tags are are tags that are relevant to concepts or code that appears directly in the question. Tags were selected based on the tagging system on the Stack Overflow website. Participants were also instructed that they were allowed to suggest tags that did not appear in the list.

The tasks were selected based on a defined list of criteria. The 9 tasks were broken down into 3 categories, 3 tasks per category. These categories were defined based on personal experience as well as research. The challenge here was creating a sequence that progressed in difficulty for each participant. Because each person has different experience, certain topics that some may find challenging others may find more simple. Fortunately I was able to confirm, via participant surveys, that the tasks increased in general with complexity.

The first category was Simple. The content consisted of topics taught commonly in a first year computer science course. Some examples include simple data types, operators, control structures, and basic properties of C/C++ language. The next category was Average. This was content beyond first year computer science level and comes from experience developing. Examples from the Average level include specific details of data structures and involved applications of aspects from the Simple level. Finally there was the Complex level. This category were application of more difficult or compound topics including algorithm designs, complicated memory management techniques, and obscure or intense properties of the C/C++ language.

IV. Analysis

The first step of analysis took place before the studies were conducted. AOIs had to be manually determined. The categories included: title, description of question, code, relevant tags, distractor tags, and keywords. Keywords are relevant words/concepts/pieces of code that directly relate to tags and can be found throughout the postings.

The first analysis done was tag accuracy. Tag accuracy was defined upon relevant tags chosen versus distractor tags chosen. The average accuracy

among participants was 90.57%, ranging from about 81% up to 100%. The average amount of tags selected per tags were 3. Feedback of overall confidence generally reflected how well the participant did. For example, a participant that did better reported higher confidence. Then accuracy was was split up by category level. For the Simple level accuracy was at 97.46%, Average at 89.76%, and Complex was 87.04%. So a trend of tag accuracy decreasing with difficulty was determined.

a. Fixation Count & Duration

On taking an average of fixation duration it was found that most fixation time was spent on code (28.92%) and then description (34.18%). The least amount of time was on title with 6.19%. It was also found that between distractor tags and relevant tags there were approximately equal fixation times, 15.28% and 15.44% respectfully.

Again, data was split up among task categories for analyzing. It was found that tags and description were mostly consistent across categories. In splitting up the data two trends were determined. As complexity increases in tag categories it was found that developers spend more time on code and less fixations on title.

Looking at fixation count it was determined that the distributions were similar to those in duration with 26.26% on code, 39.28% on description, 11.99% on relevant tags, 12.23% on distractor tags, and 7.24% on title. So again there is most fixations on code and description, about equal counts on either type of tag, and least fixation on title. In splitting up category difficulty the trends appeared again of increase of fixation on code and decrease of fixation on title with increasing complexity.

b. Non-Novice v. Novice

The next step of analyzing was further breaking down data, now by participant. The participants were split into two groups: novice and non-novice. The novice category consisted of non-Computer Science majors or freshman who had a year or less experience in developing. The non-novice group consisted of upperclassmen Computer Science majors with several years experience developing (averaging about 3-5). On average non-novice performed slightly better than novice in regards to tag accuracy with 91.92% and novice at 87.21%. Where the novice category was able to perform better were tasks in the average category but they also selected only 1-2 tags on average giving them a slight advantage. The average tag suggestions per task was 3-4 for non-novice and 2 tags for non novice. This leads to an assumption that non-novice were both more confident and were also able to better understand concepts in order to assign more tags.

In regards to fixation count and duration, the trends of increasing fixations on code and decreasing fixations on title were persistent with non-novice developers but were not present in the novice participants. Another clear result was that non-novice developers were more balanced in their fixation distributions over code, and title and description. On the other hand novice developers were having far less fixations on code and spent much more time on title and description. The average outcomes for distribution were as follows:

Non-Novice	Novice
Fixation Duration Code: 32% Title & Description: 37%	*Fixation Duration* Code: 22% Title & Description: 46%
Fixation Count Code: 32% Title & Description: 43%	*Fixation Count* Code: 24% Title & Description: 50%

c. Keywords

Keywords are words, phrases, or code features that directly appear in the postings and relate to tags. For the purposes of this study keywords were manually defined as areas of interest. The end goal here might be to use eye-tracking and other techniques to automatically identify keywords based on user fixations.

The first analysis taken into consideration for keywords was first time to fixation, that is the first time a user fixates on a certain area of interest. The outcome was 3.14 seconds to title, .99 seconds to

fixation, 9.85 seconds to code, 33.53 seconds to distractor tags, 33.17 seconds to relevant tags, and 1.65 seconds to keywords. The first inference taken from this is that users fixate on tags last. What this means is that they fully evaluate the posting before taking any tags into consideration, depicting a sequential reading pattern. It was also realized just how quickly a participant focuses on a keyword, averaging under 2 seconds. This means users generally are able to find important posting features quite quickly.

Then I evaluated visits. Visits are how many times a user enters an AOI, then leaves and comes back. I found that keyword features throughout the text had a much higher visit rate than other other feature. On average keywords get about 20 visits per posting. The feature with the next highest visit rates are tags averaging around 10-11 visits. I then looked at average fixations for keywords. Users spent about 26% of fixations (both count and duration) on keywords. Considering the small amount of visual space a keyword occupies on the screen, this is quite significant.

V. Conclusion

The most obvious conclusion that can be drawn from the data is that fixation count and fixation correlation distributions are often quite similar. Additionally, users also spent average time considering relevant and distractor tags. An obvious trend that appeared in the data was that with an increase in task complexity, there was an increase of fixations on code and a decrease of fixations on title. The two trends were especially true for non-novice developers.

Non-novice developers were able to perform better in regards to assigning tags. On average they assigned more tags which reflects on their confidence level and also their ability to better recognize concepts and what is occurring in the code. Non-novice developers also depended more on code as tasks increased in complexity which shows that with more experience developing they were able to depend more on code to assign tags because they have a deeper understanding.

Novice developers had less accuracy in tag assignment when comparing against the non-novice developers. They assigned less tags on average which reflects on lower confidence levels and their inability to assign as many tags. In addition, as task complexity increased they depended more on title and description to assign tags which leads to the assumption that they were unable to understand the code as well and depended on plain text to assign tags.

From a visual and statistical analysis (first time to fixation), it was clear that all the participants in this study had a sequential reading pattern. From previous studies [4] it is clear that different types of reading patterns and styles can affect outcomes such as efficiency and accuracy. An example is a reader who evaluates their answer (in this case tags) and then reads through the content to make their selection. Since this did not happen here there was not much to do in regards to comparing accuracy among developers with different reading patterns.

Keywords were visited early and re-visited often throughout tag evaluation. This can lead us into a weighting system which allows us to give preference to keywords that earn more attention. The keywords with higher weights can eventually be used for tag prediction.

VI. Future Work

The future of this study is to continue on with tag prediction. Now that studies have been conducted on how developers read and comprehend code, this gaze data (and hopefully more that will be gathered) can be used to train machine learning algorithms. The current algorithms being considered for the future of this study include linear support vector machines (SVM), Naive Bayes, and Random Forest. It is worth mentioning that all three studies [1]-[3] mentioned in Related Work used SVM and versions of Naive Bayes in their work predicting tags.

There will also need to be some further development in regards to keywords. It will be important to identify keywords in text automatically (versus manually defining) in order to aid in tag prediction. It is worth considering existing tag prediction studies and compounding this with eye

tracking techniques. It will always be important to recognize code features for relevant keywords. For example, a system working with C++ would need to consider "*" as a relevant keyword to the tag "pointer". This feature will have to also recognize and span across languages if it steps outside of a single language.

References

[1] A. K. Saha, R. K. Saha, and K. A. Schneider, "A discriminative model approach for suggesting tags automatically for stack overflow questions," in Proceedings of the 10th Working Conference on Mining Software Repositories , 2013.

[2] C. Stanley and M. D. Byrne, "Predicting tags for stackoverflow posts," in Proceedings of ICCM , 2013, vol. 2013.

[3] S. Schuster, W Zhu, Y. Cheng, "Predicting Tags for Stack Overflow Questions", 2013.

[4] A. Goswami, G. Walia, M. McCourt, G. Padmanabhan, "Using Eye Tracking to Investigate Reading Patterns and Learning Styles of Software Requirement Inspectors to Enhance Inspection Team Outcome", in Proceedings of ESEM, 2016.