Improving Stack Overflow Tag Prediction Using Eye Tracking

Alina Lazar, **Youngstown State University**
Bonita Sharif, Youngstown State University
Jenna Wise, Youngstown State University
Alyssa Pawluk, Youngstown State University
Ali Morris, Youngstown State University

I) Goals and Purpose

Software developers use Stack Overflow to post questions and answers related to programming and computer science problems they need to solve. Questions such as seeking input on some efficient and time-saving methods of coding a particular program, getting help on solving various bottlenecks in coding are commonly seen. When users submit questions on Stack Overflow they need to submit at least one and up to five tags in addition to their question (see Figure 1). These tags attached to each question broadly identify the programming language talked about, the problem type in discussion and maybe some other fine grained categories the question belongs to. The tags associated with each question help with information retrieval or user queries.

The goal of this project was to develop a tag prediction system utilizing eye tracking that will improve the accuracy of auto-generated Stack Overflow question tags. These Stack Overflow tags are important because they allow users to be able to further depict a problem within a program, or to precisely answer a programming question when users are on the Stack Overflow network. The main research question for our project is as follows:

- To what degree do programmers focus on the keywords that tag extraction techniques generate?

Based on the results of the previous question, another follow up study can be done to address the following two questions. In this project however, we only focused on the previous question above.

- To what degree do the top n keywords from our approach and the standard approach match our Oracle generated keywords?
- What are the best machine learning algorithms that can be successfully used to make such predictions?

II) Related Work

When users submit questions on Stack Overflow they need to submit at least one and up to five tags in addition to their question. These tags attached to each question broadly identify the programming language talked about, the problem type in discussion and maybe some other fine grained categories the question belongs to. The tags associated to each question help with information retrieval or user queries. For example, it may be very useful when users try to identify duplicate questions or related questions to a particular problem. Several approaches [1]–[4] for automatically generating tags from short questions or text have been recently developed. The Stack Overflow dataset is perfect for this problem as it provides the ground truth (as the author of the question adds these tags). The problem with this approach is that this multi-label, multi-class classification solution does not provide very good accuracy. The best accuracy reported is 68.47% [1].

Though the studies were not identical, they mostly consisted of data mining many postings from a dataset to extract important features and tags. Machine learning algorithms were trained to recognize these features and corresponding tags. When new postings are created their tags are assigned based on previous posts that have similarities [1]-[4]. The problem with this approach is that this multi-label, multi-class classification solution does not provide very good accuracy.

An automatic tag assignment system of questions on Stack Overflow [3] inspired us to use our training data set from an online competition, where text skills were tested such that participants had to predict tags based on body text using big data sets. Each of these studies uses a form of data mining from a large data set to reveal necessary text information to be used for tag prediction.

While these studies are useful, the accuracy can be improved using eye-tracking technologies.

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, pp. 73–76.

[2] M. Lipczak and E. Milios, "Learning in efficient tag recommendation," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 167–174.

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

[4] S. Beyer and M. Pinzger, "Synonym suggestion for tags on stack overflow," in *Proceedings of the 2015 IEEE 23rd International Conference on Program Comprehension*, 2015, pp. 94–103.

[5] 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 the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2016, pp. 1-10

III) **Process**

This project focuses 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.

First, we generated 9 tasks from the Stack Overflow database, all consisting of C++ questions that ranged in difficulty starting from simple to more complex, and we each generated a list of 10 tags per task (5 relevant, 5 distractors). Next, the study was conducted on a total of 15 participants from Youngstown State University consisting of electrical engineers, computer science and computer information systems majors using our Software Engineering and Empirical Studies Lab, where we asked participants to associate up to 5 tags for each task. There, we could record participant gaze data, our feature set, like fixations (eye movement counts on screen), duration (total amount of fixation time), saccades (quick eye movements between fixations) and areas of interest (specific areas on screen where eye movement is captured). The areas of interest consisted of:

- Question title,
- Question description,
- Code,
- Relevant tags,
- Distractor tags
- Keywords — relevant content in question that we thought were important to tag the question.

The keywords were picked from the question body and code (if available in the question body) by two authors of this report separately and were double checked together to come to a consensus.

See the figure below for how a question looked like as presented to the participants.

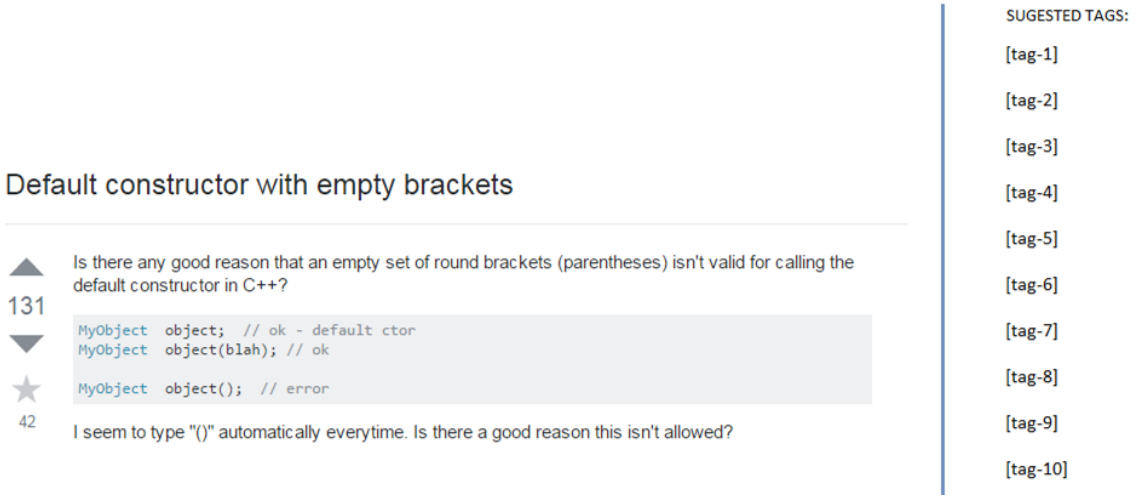

Figure 1: Sample Task Representation

We used the eye tracking data to determine weights for keywords that are read within the question. We also included other characteristics from the eye tracking data such as regressions, text and source code element visited, and sequence of visits in time. We then analyzed the data to determine where their focus was concentrated and used it to extract key words for our prediction system.

The training and testing sets used were from the online Kaggle competition Facebook Recruiting III - Keyword Extraction. This data was pulled from an existing Stack Overflow data dump, both consisting of 13.6 GB. As a preprocessing step, we applied TF-IDF (Term Frequency-Inverse Document Frequency) to the question's title and body to extract a high dimensional feature vector. Next, we trained a binary classification algorithm using the one-vs-rest approach to predict each tag value. This means building a model for the specific tag "c#", and mark the questions positive if their tags included "c#", negative otherwise.

IV) **Results and Discussion**

We now present the results of the eye tracking study. Figure 2 below shows the areas of interest in the questions we asked to the participants.

The first analysis done was tag accuracy. Tag accuracy was defined upon relevant tags chosen versus distractor tags chosen. The average accuracy (See Figure 3) among

participants was 90.57%, ranging from about 81% up to 100%. The average amount of tags selected per question 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 split up by difficulty 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.

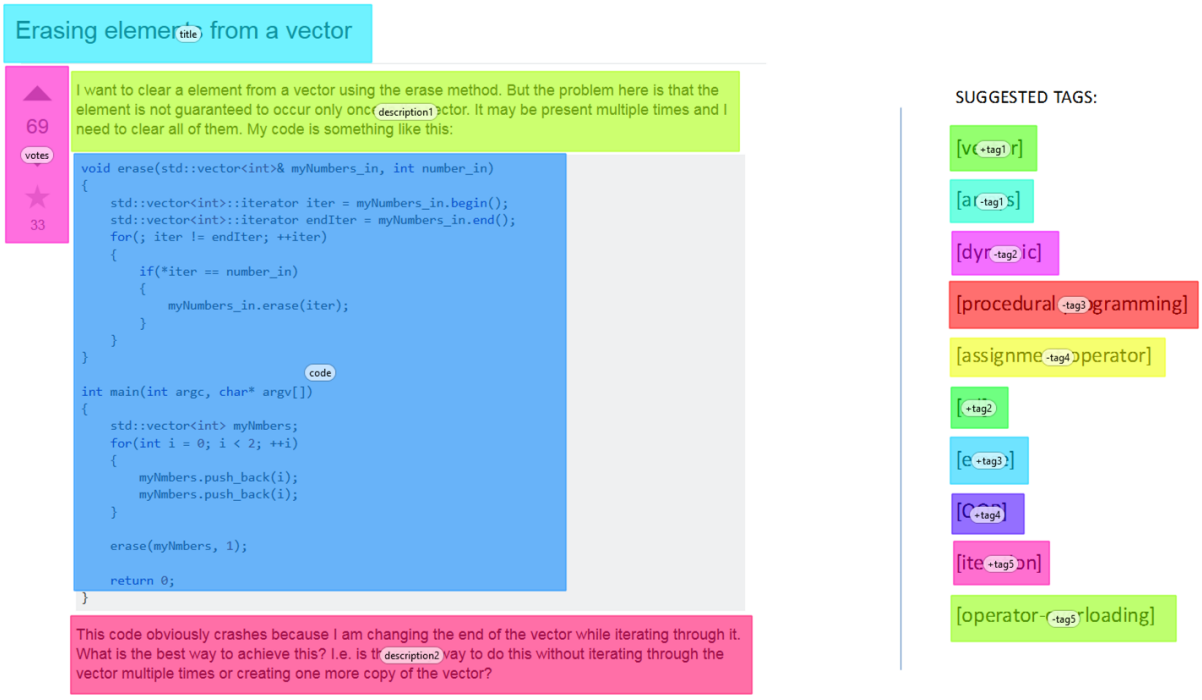

Figure 2: Areas of Interest on the Stack Overflow question

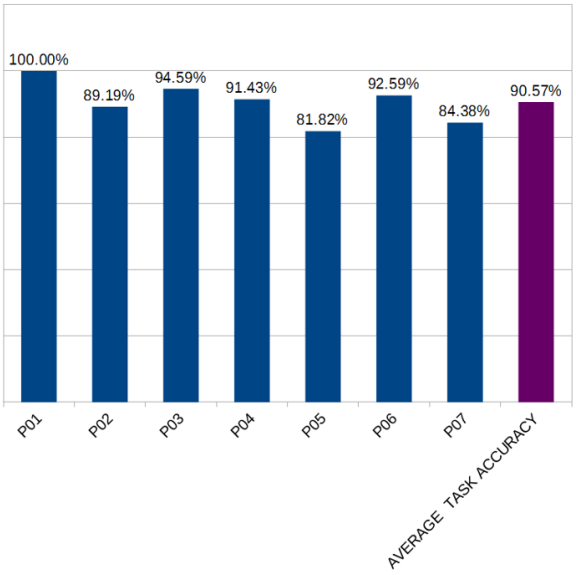

Figure 3: Tag Assignment Accuracy by Participant

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 (See Figure 4). 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.

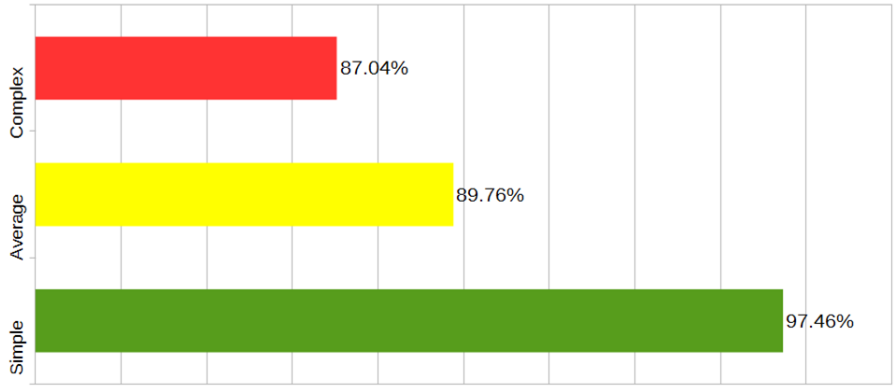

Figure 4: Tag Assignment Accuracy by Difficulty

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 (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 (See Figures 5-7), 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%

Figure 5: Distribution of fixation counts and duration.

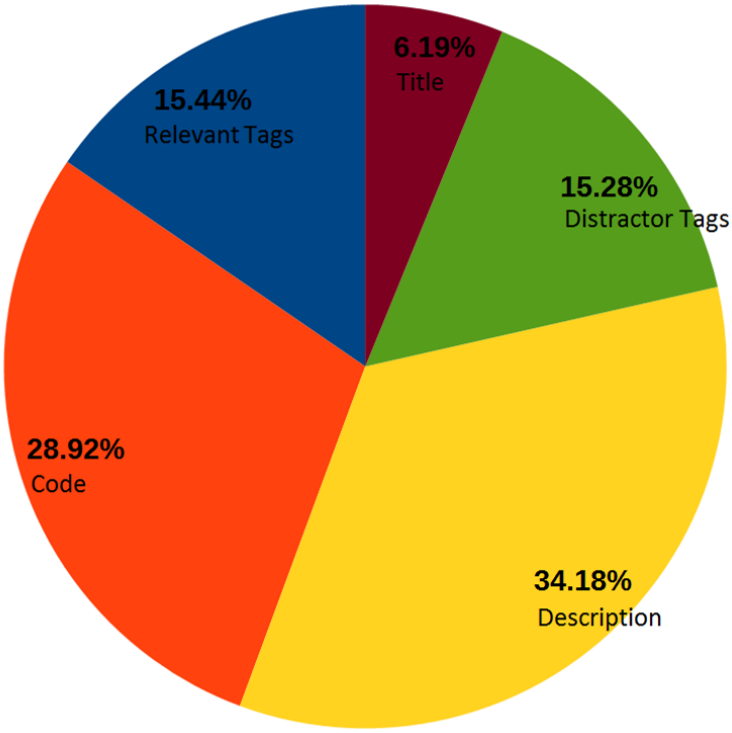

Figure 6: Overall Fixation Duration. Distribution was similar for Fixation Counts. Averages of all participants shown.

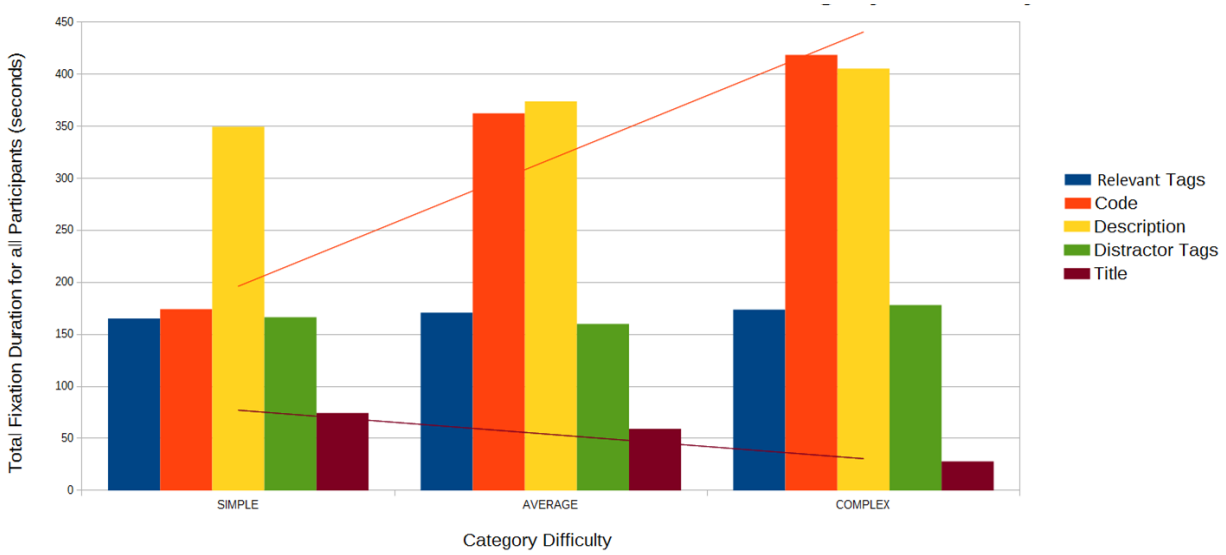

Figure 7: Overall fixation duration over categories

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 figure below shows keywords for one task. See Figure 8 for an example of keywords chosen.

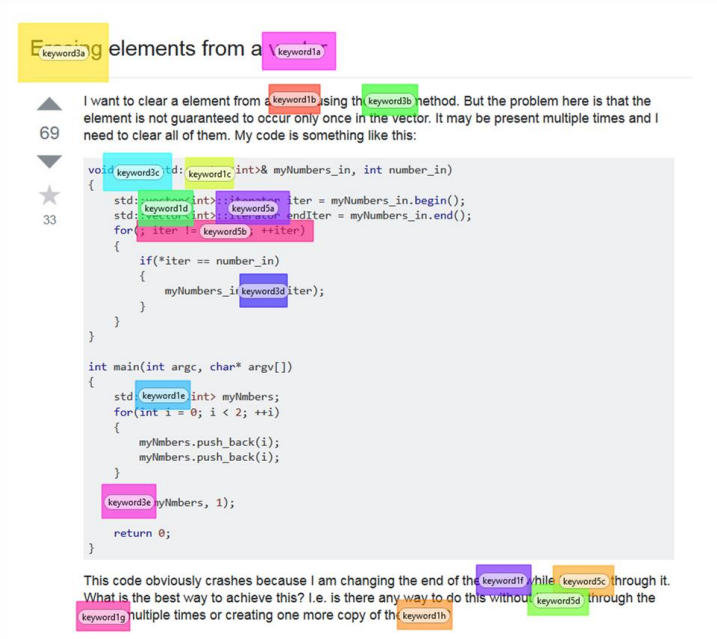

Figure 8: Keywords chosen.

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. See Figure 9. 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 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.

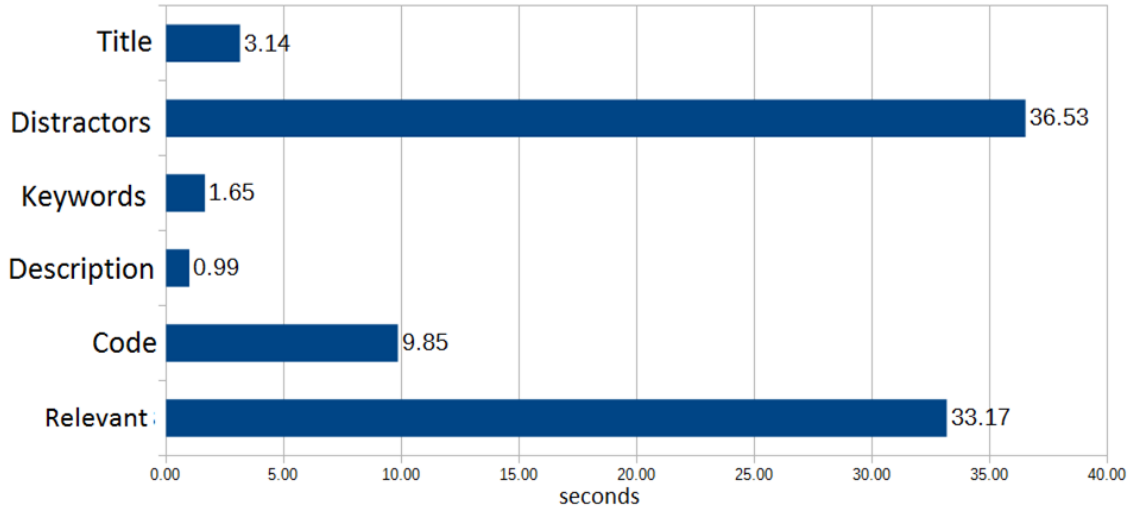

Figure 9: Time to First Fixation showing quick fixations on keywords

Our results showed trends in the data. We developed a tag score (amount of relevant vs. distractor tags chosen) for all participants. Three tags were selected on average and there was an average tag score of 90.57% overall. Once our data was split, we found certain data trends. We found that as task complexity increased, developers spent more time on code and less time looking at title. The highest number of fixations were on code, description and tags, and the least fixations on title. We further divided our results when we broke our participants into two categories: novices and non-novices.

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 [5] 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.

V) **Future Work**

There are many possible opportunities 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]-[4] 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.

In summary, we plan to use our developed keyword lists to further inform our Stack Overflow tags. We concluded that keywords were visited early and re-visited throughout the study so that those with high weights can be used for eventual tag prediction. We also hope that we can use these keywords to recognize specific code features across multiple programming languages.

VI) **Web Links**

http://ysu-creu2016-2017.blogspot.com/
http://creu-research.ysu.edu/

VII) Presentations and Publications

Ohio Celebration of Women in Computing (OCWIC) 2017: Saw Mill Creek Resort, Sandusky, OH, February 23-24, 2017

Youngstown State University QUEST A Forum for Student Scholarship 2017: Youngstown State University, Youngstown, OH, April 4, 2017

Choose Ohio First Research Poster Conference 2017: Cleveland State University, Cleveland OH, April 23, 2017