Seeking Programming-related Information from Large Scaled Discussion Forums, Help or Harm?

Yihan Lu
School of Computing, Informatics & Decision Systems Engineering, Arizona State University,
699 S. Mill Ave., Tempe AZ, USA
lyihan@asu.edu

I-Han Hsiao
School of Computing, Informatics & Decision Systems Engineering, Arizona State University,
699 S. Mill Ave., Tempe AZ, USA
Sharon.Hsiao@asu.edu

ABSTRACT

Online programming discussion forums have grown increasingly and have formed sizable repositories of problem solving-solutions. In this paper, we investigate programming learners' information seeking behaviors from online discussion forums. We design engines to collect students' information seeking processes, including query formulation, refinement, results examination, and reading processes. We model these behaviors and conduct sequence pattern mining. The results show that programming learners indeed seek for programming related information from discussion forums by actively searching on the site and reading posts progressively according to course schedule topics. Advanced students consistently perform query refinements, examine search results and commit to read, however, novices do not. In addition, advanced students commit to read posts, but novices only skim.

Keywords

Programming; Information Seeking; Hidden Markov Model; Discussion Forums; Sequential pattern mining;

1. INTRODUCTION

In teaching and learning programming, students are typically asked to refer to API (Application Programming Interface) or programming textbooks for relevant information (i.e. code syntax or code examples). In recent years, open & free online communities (such as homework-help sites, discussion forums for MOOCs courses etc.) have grown increasingly and have formed sizable repositories of problem solving-solutions. They are filled with thousands of programming problem-solving tips, such as "*how-to*" questions [1], people-valued examples, and the examples' explanations [2] etc. On the other hand, from a constructive point of view, the action of articulating a problem and initiating search or referencing can also be a valuable learning activity as well as browsing the solution. In software engineering field, such programming information seeking has already been recognized as a core sub-task in software maintenance [3, 4]. Programmers are even being referred as task-oriented information seekers, which they focus on finding the answers they need to complete a task using a variety of information sources [5]. There are tools that have been built to make completing programming tasks easier, such as Mica [6]. However, none of these tools focuses on amplifying learning opportunities if any, rather, centers on task-oriented problem solving facilitation.

In addition, according to Information Foraging theory [7], finding information is human nature. To successfully form information seeking criteria for a given programming problem requires complex cognitive activities (i.e. defining and verbalizing the programming problem; refining query criteria and selecting results; strategies application etc.) To better support information seeking and learning, we focus on learners' behaviors in seeking programming-related information. Specifically, we investigate in an online large-scale discussion forum, StackOverflow, which is one of the biggest online programming Q&A sites communities and currently hosts a massive amount of heterogeneous definitions, solutions and examples of programming languages. Are those assorted content in the forum helpful or harmful for programming learners?

Studies have shown that while there is a positive connection between the usage of StackOverflow and GitHub (open source code management service), StackOverflow's users consider the site to be more attractive and beneficial for learning programming [8]. In recent learning science literature, learning-from-observing paradigm appears to be a promising strategy, which passive participants (such as lurkers who consume content without contributions) can still learn by reading the postings-and-replies exchanges from others due to the constructive responses in the content [9]. Knowledgeable students can benefit from text with cohesive gaps by making active retrieval and inferences [10]. They can also benefit from building memory and fluency through the active retrieval opportunities and to refine the conditions of application through feedback on incorrect solution attempts in problem solving [11]. On the other hand, novices may benefit from seeing examples of solution steps and from seeing the entire solution structure to make sense of the role of each step in order to construct integrated knowledge components for generating plans and sub goals [12]. In this work, our goal is to investigate what are programming learners' tactics in searching for relevant information from online discussion forums and how do they look for relevant learning materials from massive forum posts.

In this paper, we design engines to capture programming learners' activities on StackOverflow site, such as problem verbalization in queries, query revision and other information seeking processes. We collect a semester long of *informal* programming learning activities from programming discussion forum. We model their information seeking activities by using Hidden Markov Model and data mine the post of their readings.

2. LITERATURE REVIEW

2.1 Modeling Information Seeking In Learning

Traditionally, information seeking is associated with behavioral science theories, which focus on seekers' information needs, searching strategies, and how they use the information. For example, self-awareness of one's information needs, self-regulated learning strategies, information searching experience and ability, etc.[13-15]. Puustinen and Rouet [13] further classified help-seeking behavior into different types on a help-seeking continuum, a function of the helpers' capacity to adapt answers to their needs. In more recent information seeking literature, we see studies show that users commonly exhibit exploratory behavior in a great extent when performing searches [14]. Marchionini [15] identifies a range of search activities that differentiate exploratory search from look up search (i.e. fact-finding retrieval). Such behavior is especially pertinent to learning and investigating activities, which is the targeted area of interest in our research.

2.2 Modeling Learning From Discussion Forums

Over the decades, data mining on discussion forums has been carried out through various formats, network analyses, topical analyses, interactive explorers, knowledge extraction, etc. [16-18]. Due to calculation complexities (since linguistic features rely on computer processing power), most of these in-depth analyses were performed offline [19, 20]. As a result, the lesson learned could only be applied in the next iteration of system development. Recently, however, we begin to see some studies that focus on dynamic support for users [21]. With the rapid growth of free, open, and large user-based online discussion forums, it is essential, therefore, for education researchers to pay more attention to emerging technologies that facilitate learning in cyberspace. For instance, Wise, Speer, Marbouti, and Hsiao [22] studied an invisible behavior (listening behavior) in online discussions, where the participants are students in a classroom instructed to discuss tasks on the platform; van de Sande & Leinhard [23] investigated online tutoring forums for homework help, making observations on the participation patterns and the pedagogical quality of the content; Hanrahan, Convertino & Nelson [24] and Posnett, Warburg, Devanbu, & Filkov [25] studied expertise modeling in a similar sort of discussion environment.

3. METHODOLOGY

3.1 Research Platform & Data Collection

In this project, we deployed a Chrome browser plugin to track users' query, searching, and reading behaviors on StackOverflow (SO). User can search query on StackOverflow and identify their intention with this tool. The browser plugin has two main features. (1) It provides a direct search channel for users to issue queries on StackOverflow; (2) It displays users' search histories. We collect not only users' search queries, but also their search intentions, including "Knowledge seeking", "Method learning", "Problem solving", and "Other" (indicated by the user). Most importantly, we log all the users' behaviors, comprising of scrolls, clicks, selections, and corresponding actions' time. The behavior tracking function resides on StackOverflow site once initial log in via the SO search tool. In another word, all students' behaviors on StackOverflow site will be logged after at least one time log in via SO Search Tool. However, since they issue the queries directly from StackOverflow site, their intention will be marked as "not specified".

3.2 Study Setup

In order to understand the students' information seeking behaviors on discussion forums, we conducted a user study in a programming class in Arizona State University. Students were encouraged to install the browser plugin search tool. They were told that their search activities would be collected via the tool. All students' programming information seeking behavior was logged during the entire semester. Additionally, we also conducted a controlled session of lab class during the semester. In the lab class, students were instructed to solve a complex task (implement a 3-way merge sort algorithm) by using the information-seeking tool within 75 minutes. All the students' searching and reading behaviors on StackOverflow were recorded.

Students were given a pretest to examine their pre knowledge about programming. In this study, the students are split into two groups (*Novice* & *Advanced*) based on their pretest median score, which is ranged from 0 to maximum score 20.

3.3 Data Descriptive

Among 86 students in the Object-Oriented Programming class, 71 students voluntarily installed our search plugin, whose operations on SO were automatically recorded, 55 of them also used the plugin to search queries. There were 44 of them took the pretest. According to their pretest score distribution, 24 of them were identified as novices, and 20 were classified as advanced students.

3.3.1 Query data log

For these 55 students provided query information, the average query number is 9.55 (max 56, min 1, median 8), and the average number of operations is 7179 (min 1, median 2917, max 140300). In terms of the query content, the average number of words in each query is 3.76, and the number of distinct words is 573. The frequency distribution for each word approximately follows Zipf's law, which states that the relation between the word frequency and its rank is exponential in general. Considering the pre knowledge of students, queries are separate by whether the provider is novice or advanced student. The novices provided more query in average (13.2±11.7) than advanced students (8.9±9.0), but novices' length of each query (3.47±2.01) is shorter than advanced ones (4.62±2.61), which indicated a lower quality according to Belkin's research [28].

3.3.2 Operation data log

There are 466,659 operations logged including *scroll up*, *scroll down, click* and *select* for both searching and reading phases. We found that for both groups of students, novices and advanced students, generated the majority of the operations in reading and in scrolling down. There were 19.3% operations are scrolling up in the searching phase in general, which was not a trivia finding, It showed that users were going back and forward to review the posts content before they decide to click in to proceed further reading in detail, However, ideally a successful search process is that after entering the query, the best item would be shown in the first place of the search result, so that the user would not even need to scroll before clicking to view a result. However in reality, users need to scroll down when they do not feel satisfied with the results provided in the first view, and this unsatisfying ratio is reflected by the scrolling back and forward operation percentage.

On the other hand, the time cost before each operation shows that when browsing search results, users appear to spend more time (37.8%) before clicking or selecting, while they are faster when reading a specific question-answer thread. This fact indicates that users would read more carefully, or be more serious when choosing a thread to read among the search results.

Considering pre knowledge difference, the ratio of scroll back for novices were lower in searching phase compared to the advanced students, but their scroll back ratio is higher in reading phase. This indicates that the novices were more likely to make a choice without browsing more search results, and they had to read the content for more times compare to advanced students.

3.4 Programming Information Seeking Actions

In order to analyze students programming information seeking behavior on discussion forums, we categorize their actions into 6 categories based on Marchionini's [18] information seeking processes: formulate queries, query refinement, results examination, and reading. According to the amount of operations made on each single page, we further split search and reading (by median) in large-search (LS), small-search (SS), large-read (LR), small-read (SR). Table 1 describes detail of user search actions.

Based on the operation data collection and the above action definitions, 2681 actions were identified in total, and the distribution of action distribution is shown in Figure 2.

Table 1. Programming information seeking actions

Actions	Description
Query (Q)	a student issues an query to look for information from programming discussion forum
Refine query (q)	a student modifies the original Q and issues a similar query (word adjacent distance less than 0.3)
Large search (LS)	A student browses the search result page and did operations more than the median of all search pages (31 operations)
Small search (SS)	A student browses the search result page and did operations less than the median
Large read (LR)	A student reads a Q&A thread page, and did operations more than the median of all reading pages (64 operations)
Small read (SR)	A student reads a Q&A thread page, and did operations less than the median

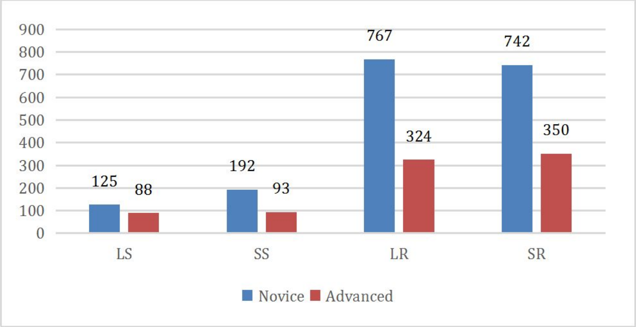

Figure 2. Number of actions identified for novices and advanced students

3.5 Modeling Programming Information Seeking From Discussion Forums Using HMM

The Hidden Markov Model (HMM) is a popular method for modeling sequential data. Previous studies have already shown its ability in modeling user information search process [26], survey design [27] and student learning process [28]. In this study, we employ the HMM to model users' hidden tactics in searching for programming related information on discussion forums, and refer the actions on the site (e.g. query refinement, results examination, content reading, information extraction) as the generated hidden tactics. The hidden tactics can be explained as the strategy used as informal learning activities by looking for programming related information.

We have a sequence of information seeking behaviors from T1 to TM, and each state is one of those predefined information seeking actions: TS = {Q, q, LS, SS, LR and SR}. HMM assumes that we also have a sequence of hidden states, from H1 to HM, and each answer type is generated by a corresponding hidden state, but different answer types can be generated by the same hidden state with different probabilities. A HMM model has several parameters: the number of hidden states HS, the start probability of each states π, the transition probabilities among any two hidden states A_{ij}, and the emission probability from each state to each action b_{ij}. By only defining the HS and π, a Baum-Welch algorithm [29] can be used to learn the emission and transition probabilities.

4. EVALUATION RESULTS

4.1 Mapping HMM Patterns to Information Seeking Processes

In this section HMM is used to detect the students' information seeking behavior pattern. In order to identify the complete sequence of information seeking operations, we only included those operations following a query recorded. The web paged that the students searched from other search engines, where queries were not included, are excluded.

The first step of using HMM is to determine the number of hidden states. A larger number of states will help to describe the model more precisely, while the risk of over-fitting is also increased. In model selection, the information criterion such as the Akaike Information Criterion (AIC) or its variants Bayesian information criterion (BIC) [29] can be used to determining the optimal number of states. Based on models best performance by AIC, we choose HS=3 and HS=5 for *Advanced* and *Novice* groups accordingly (Figure 3).

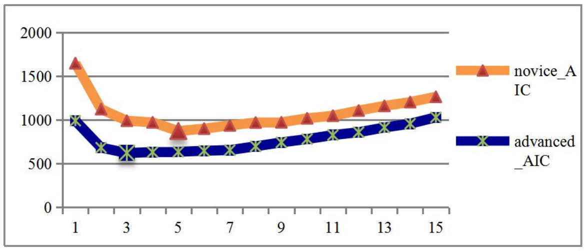

Figure 3. Choosing number of hidden state using AIC.

The emission probability of each hidden state to information seeking operations is shown in Table 2, in which the probabilities under 0.05 were removed for better presentation of the results. The hidden states can be treated as the underlying "tactics" or "principles" when students look for programming information from the discussion forum. For example, *Advanced* group HS2 demonstrates the stronger students' reading behaviors, which they appear to do more careful readings and fast browsing; while in *Novice* group HS3, students tend to perform more superficial reading than careful reading. While *advanced* group shows more coherent searching, browsing and reading behaviors (each behavior is observed by single state), novices show duo searching and browsing behaviors. *Novice* HS4 and HS1 states seem to have similar searching and browsing behaviors as *advanced* group. However, *Novice* HS5 exhibits more distinct searches by issuing queries and lower probability in refining queries. In addition, Novice HS2 shows high probabilities in small search, which can be interpreted as careless results examination.

Table 2. The hidden states of programming information seeking operations (b_{ij})

hidden states		Q	q	LS	SS	LR	SR
	HS1	0	0	0.39	0.61	0	0
Advanced	HS2	0	0	0	0	0.79	0.22
	HS3	0.76	0.24	0	0	0	0
	HS1	0	0	0.36	0.64	0	0
	HS2	0	0	0.05	0.95	0	0
Novice	HS3	0	0	0	0	0.35	0.65
	HS4	0.73	0.27	0	0	0	0
	HS5	0.85	0.15	0	0	0	0

Figure 4 is plotted according to the transition probability, and the prior probability is shown in Table 3. The probabilities under 0.05 are removed. HS3 has the highest prior probability (start probability) in *advanced* group, which means that advanced students always begin with issuing query and modifying the query. So do the majority of the weaker students. In addition, HS5 state is also another beginning state with high probability for novices. It shows that there is also a great probability that novices start issuing queries with minimal query refinement. However, what are the impacts of the amount of query refinement? We have to look at what is happening next. According to Figure 4, the *Advanced* & *Novice* state transition diagrams, there are several findings listed below:

Table 3. The prior probability of each hidden state (π)

	HS1	HS2	HS3	HS4	HS5
Advanced	0	0	**1**	-	-
Novice	0	0	0	**0.536**	**0.464**

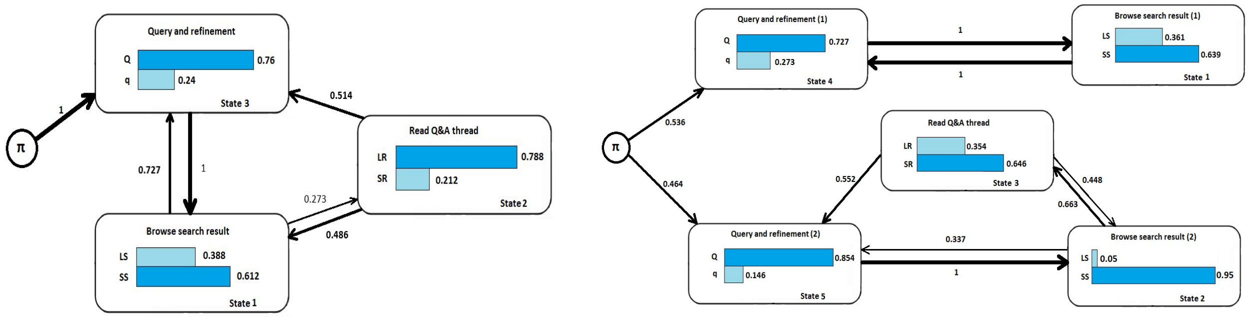

Figure 4. *Advanced* (left) and *Novice* (right) students' information seeking transition probability diagrams

4.1.1 Advanced students refine query; novices don't

Advanced students consistently performed query refinements (3:1 ratio) before they examine the results (HS3 → HS1). Novices behaved differently. Part of them followed the similar pattern as *Advanced* students did, tuning the queries before examine the results (HS4 → HS1). However, when these novices refined queries, there were no consecutive actions followed in the next step (Figure 4 – right top), which indicated that they did not go to any reading page. On the other hand, when novices did minimum query refinements (HS5 → HS2), they did manage to proceed to next step, which was the reading phase (HS5 → HS2 → HS3). This fact suggested that novices may lack of query-results examination ability and lead to no reading (HS4 → HS1). In addition, as the HS2 of *Novice* group shows, 95% of the likelihood that the operations were small searches, which means that novices tended not to scrutinize the search results, they only examined the results minimally, even move on to read forum posts (HS5 → HS2 → HS3). They could read whatever the discussion forum has recommended (i.e. top returned items).

In fact, Table 4 shows the total amount of time that each student spent on searching or reading pages. It is surprising to see that novices spent more than 130 minutes on just reading, while advanced students spent about 40 minutes. Similarly, novices spent more time on searching compare to advanced students. The reason of the time difference is not only they browsed more pages, but also their time spent on each page is longer. These findings indicate that the novices' searching and browsing behaviors only consist of minimum query refinement so that they had to spend more time to read and understand search results, which can be due to insufficiency of vocabulary in searching and lack of judgment in finding reading resources. We further looked into students' reading behavior and reading content in the following section. Despite the reading quality, novices' behaviors can also suggest the *hidden danger* of online large-scale discussion forums, where the existing filtering mechanisms (such as badges, acceptance, and votes) may not be enough, especially for novice learners.

Table 4. Total time spent on searching and reading average per student

total time (seconds) / student	Novice (N=24)	Advanced (N=20)
Search	340.5	146.4
Read	7870.3	2366.6

4.1.2 Advanced students read and novices skim

When students eventually landed on forum post pages and read, we found that *advanced* students committed to careful reading, while novices did more skimming (*Advanced* HS2: 0.79 LR; *Novice* HS3: 0.65 SR). In fact, we found that novices cost more time in small reading than advanced students, while in large reading advanced students spent slightly more time, but there was no significant difference between groups. These results reveal that novices performed less reading in search results filtering, but once they did, they would spend time to read. Thus, it led us to examine their learning effect. Do novices and advanced students have similar effects after reading?

4.2 Reading and Learning Effects

4.2.1 *Students read posts according to course schedule topics*

In order to understand what content were students' reading, we crawled all the posts that students read from StackOverflow, and performed text mining with MALLET[1] LDA toolkit with default α=30/N, β=0.01, *itr*=1000. We found students were reading the contents from discussion forums according to the course weekly topics, from week 1 *Java Basis* to week 9 *LinkedList*. We then used all the topic words generated from the LDA model to compute Shannon entropy score in estimating the topic focus (Figure 5). There are several interesting findings: *Advanced* students were generally more focused across all topics (smaller topic entropy), except week 4 and week 9. The effect was much more apparent in complex topics: *Recursive* (Table 5 shows the extracted topic words, which we found advanced students read posts regarding to a specific recursive implementation Fibonacci sequence, which novices did not). In week 4 and 9, advanced students were found to be less focused in terms of reading more diverse topics was due to those two weeks were exam periods. Therefore, it is understandable that students might read a wider range of topics that were covered over exam periods.

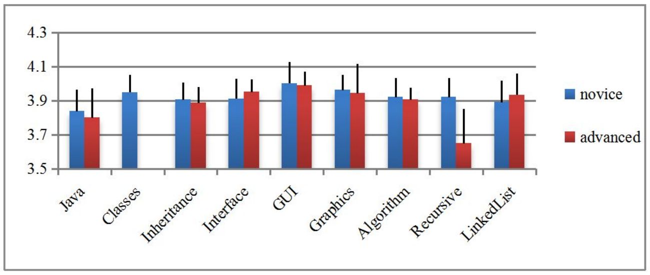

Figure 5. Weekly readings' keywords by novices and advanced students

Table 5. Recursive topic words by novices and advanced students

Novice: {type, code, recursive, dynamic, void, write, result, example, loop, print, add, wikipedia, error, int, version, method, operator, pseudo, easy, program, static, mathematics, call, line, learn, number, work, value, function, undefined}

Advanced: {function, method, value, static, return, int, change, version, recursive, result, error, mathematics, program, line, number, fibonacci, sequence, fib, wikipedia, operator, pseudo, easy, type, print, example, code, learn, void, traverse, loop}

4.2.2 *Learning Effects*

Based on the percentage of large read rate in reading pages, we found that the more students spending time in reading on StackOverflow, the higher final score they obtained (r=0.418, p<0.01). Additionally, we found that the slope of novices and advanced students had little difference, while the intercept of novices is higher. This fact indicates that novice and advanced students gained the same benefits from increasing large read rate, however, in order to achieve the same score, novices has to read more carefully. Figure 6 shows the connection between large read rate and final exam score.

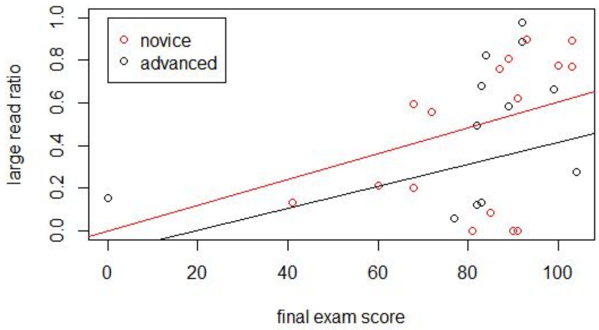

Figure 6. Final score vs. Large read rate

5. CONCLUSIONS

5.1 Summary

In this study, we designed a programming information seeking framework with a browser plugin to collect students' programming information seeking behavior data from discussion forum StackOverflow. Students' query intention, time spent and all actions were logged. We modeled programming learners' query formulation, refinement, results examination, and reading processes with Hidden Markov Model. We conducted sequence pattern mining. The results showed that programming learners indeed seek for programming related information from discussion forums by actively searching on the site and reading posts progressively according to course schedule topics.

The result of this study showed that programming novices usual spend more time in browsing search result and reading, while the sequential due to their lack of pre knowledge. As long as they can read as well as advanced students, they can learn as much as advanced students according to the learning evaluation result.

All the study results shed lights on programming learners seek for learning resources from large-scale online discussion forums. We anticipate this work serves as guidelines for educational technologists to design better effective tools to facilitate learning via programming information seeking process.

5.2 Limitations and Future Work

There are a few limitations in current study. First of all, after students log in from the browser at least once, all their activities on StackOverflow will be recorded. However, when students search from search engines (i.e. Google) and land on StackOverflow site, their initial queries will not be captured. A more completed data collection should include all queries that the students search in information seeking.

Moreover, we mainly take into account of students' query and mouse actions without considering other keystrokes' actions. Another common information seeking behavior is to use Ctrl+F on the keyboard to search keyword with in a web page, which was not captured in the study. This operation can be a convenient and fast method to locate useful information when browsing web pages, including discussion forums.

In the future, we will consider a more completed data collection and more exhaustive evaluation. Most importantly, we aim to design an adaptive programming information seeking tool to help novices effectively navigate search results.

6. REFERENCES

[1] Vasilescu, B., Serebrenik, A., Devanbu, P., & Filkov, V. (2014, February). How social Q&A sites are changing

[1] http://mallet.cs.umass.edu

knowledge sharing in open source software communities. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing* (pp. 342-354). ACM.

[2] Treude, C., O. Barzilay, and M. Storey. How do programmers ask and answer questions on the web?: NIER track. in *Software Engineering (ICSE), 2011 33rd International Conference on*. 2011.

[3] Seaman, C.B. The information gathering strategies of software maintainers. in *Software Maintenance, 2002. Proceedings. International Conference on*. 2002. IEEE.

[4] Sharif, K.Y. and J. Buckley. Developing schema for open source programmers' information-seeking. in *Information Technology, 2008. ITSim 2008. International Symposium on*. 2008. IEEE.

[5] Sim, S.E., *Supporting multiple program comprehension strategies during software maintenance*. 1998, University of Toronto.

[6] Stylos, J. and B.A. Myers. Mica: A Web-Search Tool for Finding API Components and Examples. in *Visual Languages and Human-Centric Computing, 2006. VL/HCC 2006. IEEE Symposium on*. 2006.

[7] Kuhlthau, C.C., Inside the search process: Information seeking from the user's perspective. *Journal of the American Society for Information Science*, 1991. 42(5): p. 361-371.

[8] Eickhoff, C., Teevan, J., White, R., & Dumais, S. (2014, February). Lessons from the journey: a query log analysis of within-session learning. In *Proceedings of the 7th ACM international conference on Web search and data mining* (pp. 223-232). ACM

[9] Chi, M.T.H. and R. Wylie, The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. *Educational Psychologist,* 2014. 49(4): p. 219-243.

[10] McNamara, D. S. (2001). Reading both high-coherence and low-coherence texts: Effects of text sequence and prior knowledge. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 55(1), 51.

[11] Matsuda, N., Cohen, W. W., Sewall, J., Lacerda, G., & Koedinger, K. R. (2008, June). Why tutored problem solving may be better than example study: Theoretical implications from a simulated-student study. In *Intelligent Tutoring Systems* (pp. 111-121). Springer Berlin Heidelberg.

[12] Catrambone, R. (1996). Generalizing solution procedures learned from examples. *Journal of Experimental Psychology: Learning, Memory, and Cognition,* 22(4), 1020.

[13] Puustinen, M. and J.-F. Rouet, Learning with new technologies: Help seeking and information searching revisited. *Computers & Education*, 2009. 53(4): p. 1014-1019.

[14] Zimmerman, B.J. and M.M. Pons, Development of a Structured Interview for Assessing Student Use of Self-Regulated Learning Strategies. *American Educational Research Journal*, 1986. 23(4): p. 614-628.

[15] Marchionini, G., Exploratory search: from finding to understanding. *Communications of the ACM*, 2006. 49(4): p. 41-46.

[16] Dave, K., M. Wattenberg, and M. Muller, Flash forums and forumReader: navigating a new kind of large-scale online discussion, in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. 2004, ACM: Chicago, Illinois, USA. p. 232-241.

[17] Indratmo, J. Vassileva, and C. Gutwin, Exploring blog archives with interactive visualization, in *Proceedings of the working conference on Advanced visual interfaces*. 2008, ACM: Napoli, Italy. p. 39-46.

[18] Guerra, J., Sahebi, S., Lin, Y. R., & Brusilovsky, P. (2014). The problem solving genome: Analyzing sequential patterns of student work with parameterized exercises. in *The 7th International Conference on Educational Data Mining. 2014*: London, UK.

[19] Wen, M., D. Yang, and C. Rose. Sentiment Analysis in MOOC Discussion Forums: What does it tell us? in *the 7th International Conference on Educational Data Mining*. 2014. London, UK.

[20] Wang, X., Yang, D., Wen, M., Koedinger, K., & Rosé, C. P. Investigating how student's cognitive behavior in MOOC discussion forums affect learning gains, in *The 8th International Conference on Educational Data Mining*. 2015: Madrid, Spain.

[21] Enamul Hoque, G.C., Shafiq Joty. Interactive Exploration of Asynchronous Conversations: Applying a User-Centered Approach to Design a Visual Text Analytic System. in *Workshop on Interactive Language Learning, Visualization, and Interfaces*. 2014. Baltimore, Maryland.

[22] Wise, A. F., Speer, J., Marbouti, F., & Hsiao, Y. T. (2013). Broadening the notion of participation in online discussions: examining patterns in learners' online listening behaviors. *Instructional Science*, 41(2), 323-343.

[23] Sande, C.v.d., Free, open, online, mathematics help forums: the good, the bad, and the ugly, in *Proceedings of the 9th International Conference of the Learning Sciences - Volume 1*. 2010, International Society of the Learning Sciences: Chicago, Illinois. p. 643-650.

[24] Hanrahan, B.V., G. Convertino, and L. Nelson, Modeling problem difficulty and expertise in stackoverflow, in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion*. 2012, ACM: Seattle, Washington, USA. p. 91-94.

[25] Posnett, D., Warburg, E., Devanbu, P., & Filkov, V. (2012, December). Mining stack exchange: Expertise is evident from initial contributions. In *Social Informatics (SocialInformatics), 2012 International Conference on* (pp. 199-204). IEEE.

[26] Han, S., Z. Yue, and D. He. Automatic detection of search tactic in individual information seeking: A hidden Markov model approach. in *iConference 2013*. 2013. arXiv preprint arXiv:1304.1924.

[27] Hsiao, I. H., Han, S., Malhotra, M., Chae, H. S., & Natriello, G. (2014, June). Survey sidekick: Structuring scientifically sound surveys. In *Intelligent Tutoring Systems* (pp. 516-522). Springer International Publishing.

[28] Piech, C., Sahami, M., Koller, D., Cooper, S., & Blikstein, P. (2012, February). Modeling how students learn to program. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (pp. 153-160). ACM

[29] Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics,* 41(1), 164-171.