

Personalized Information Seeking Assistant (PiSA): from programming information seeking to learning

Yihan Lu¹  · I-Han Hsiao¹

Received: 15 October 2016 / Accepted: 28 April 2017 / Published online: 9 May 2017
© Springer Science+Business Media New York 2017

Abstract Online programming discussion forums have grown increasingly and formed sizable repositories of problem-solving solutions. In this paper, we investigate programming learners' information seeking behaviors in online discussion forums, and provide visual navigational support to facilitate information seeking. We design engines to collect students' information seeking behaviors, and model these behaviors with sequence pattern mining techniques. The results show that programming learners indeed seek for information from discussion forums by actively search and read progressively according to course schedule topics. Advanced students consistently perform query refinements, examine search results and commit to read, however, novices do not. Finally, according to the lessons learned, we propose, design and evaluate Personalized Information Seeking Assistant system to help query refinement by summarizing the search results and to provide social-based browsing history. Findings suggest that paying attention to the query history may lead to further reading events, which subsequently resulting in potential learning activities.

Keywords Behavior modeling · Information seeking behavior · Computing education · Novice programming learning

1 Introduction

In the last two decades, researchers have successfully exploited information foraging theory (Pirolli and Card 1999) in designing tools to help people navigate and explore the Web, including visual information seeking, collaborative information seeking, exploratory

✉ Yihan Lu
lyihan@asu.edu

¹ Computing System & Informatics Lab, School of Computing, Informatics, and Decision System Engineering, Arizona State University, Tempe, AZ, USA

and interactive search, etc. (Chi et al. 2007; Choo et al. 2000; Evans and Chi 2008; Wildemuth and Freund 2012). Krikelas (1983, p. 7) information seeking behavior model tells us “information seeking begins when someone perceives that the current state of knowledge is less than that needed to deal with some issue (or problem). The process ends when that perception no longer exists.” This emphasizes the seekers’ state of knowledge and their initiated efforts.

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. (Bilal 2002; Puustinen and Rouet 2009). Puustinen and Rouet (2009) 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 (Hearst 2009; Teevan et al. 2004; White and Drucker 2007; Wilson and Schraefel 2008). Marchionini (2006) 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.

In the context of information seeking for programming, we can see a great deal of tools are designed mainly to extract relevant information from the web to assist in current coding tasks and save time spent navigating through codes when gathering information (Ko et al. 2006). These tools include navigational shortcuts to the code in Integrated Development Environment (IDE) (Singer et al. 2005), leveraging version history data to better use API (Stylos and Myers 2006) and integration of web search or recommending source code examples in developing environment (Brandt 2010; Holmes and Murphy 2005; Hsiao et al. 2008; Stylos and Myers 2006). However, these systems were designed mainly to extract relevant information from the web to aid in current coding tasks and save time that would otherwise be spent navigating through codes to gather information. Moreover, with the rise of Web 2.0, we also see that a variety of technologies (blogs, tags, wikis, recommenders etc.) are emerging to exploit social information foraging, such as online collaborative programming (social coding in GitHub¹), Q&A websites, crowd-sourcing suggestions, etc. (Dabbish et al. 2012; Hsiao et al. 2008; Mujumdar et al. 2011; Nasehi et al. 2012; Treude et al. 2011; Vasilescu et al. 2014) However, almost all of these aforementioned technologies are targeted at problem-solving augmentation, reducing coding cognitive overhead when coding, and utility features enhancement (i.e. collaboration) for experienced programmers. Tools designed to support learning programming during search exploration are less emphasized.

Thus, in this work we focus on researching the issues that how do students learn programming during search. The contribution includes two parts: first we investigate in how do novice students look for programming-related information from large-scaled discussion forum; Then we summarize two classroom studies in the following structure: we firstly review adjacent related work from information seeking to learning, programming learning modeling and modeling learning from online discussions. Following the related work, we lay out the overall methodology that we applied to uncover programming information seeking strategies and the first study results. We then elaborate the second study with the proposed research platform, Personalized Information Seeking Assistant (PiSA). Finally, we present the second study evaluation results and discuss the educational implications.

¹ github.com: open source code management service.

2 Literature review

2.1 Search behavioral modeling

Search engine user behavior modeling has been studied for years to understand the preference of web search users. In these studies, a user model is a set of rules that allow us to simulate user behavior on a search engine result page in the form of a random process (Cole et al. 2011). These studies also discuss different bias affecting the models, for example position bias means the first link listed in search result has a higher probability to be clicked (Cutrell and Guan 2007; Joachims et al. 2005; Lorigo et al. 2008), while Kiseleva et al. (2015) reported that user with expertise manage to detect better answers as they dig them from the bottom of search result. User models, especially click model during web search, helps to detect general preference of users. However, it takes little account of the text content in search results, and the actual need of users is difficult to collect when they use search engine. Ageev et al. (2011) proposed a method analyzing searcher success in relation to the searcher behavior with realistic search tasks. Additionally, when users use search engine just for learning, it is interesting to study the modeling difference. Data mining techniques can be involved to study the user behavior patterns.

For specific programming learning behavior, sequential pattern mining techniques has been applied in several studies, such as programming problem solving (Guerra et al. 2014), programming assignments progression (Piech et al. 2012), learning programming with dialogic tutor (Boyer et al. 2011). Beal et al. (2007) studied about modeling engagement level of students by analyzing their action traces on a tutoring system with HMM. Jeong et al. (2008) study a computer agent was taught by students, and the student's behavior in learning was captured with HMM.

These studies proved that students do have different behavior patterns in learning. However, it is still not answered that what is the connection between learning behavior and learning affect behavior, considering the students' knowledge background. Reinecke et al. (2013) studied how users judge a design of a website by colorfulness and visual complexity, and modeled their evaluation with quad-tree and R-tree. In another study, a visualization system was designed to help learners understand their learning progress and helps to provide optional service. Additionally, interactive visualization is found to improves students' learning by engaging them to interact with their learners' models (Hsiao et al. 2013; Bull et al. 2016).

2.2 Linkages from information seeking to learning theory

From behavior sciences to learning sciences, we have identified (a) help seeking, (b) open student modeling, and (c) concept mapping literature are close related to the targeted research of interests in supporting self-awareness of one's information needs and supporting self-regulated learning.

2.2.1 Help-seeking in learning

From theoretical perspective, looking for information is a means to complement current knowledge and cognitive skill acquisition action (Aleven et al. 2016). Empirical study results showed that learners often use help systems ineffectively or ignore them altogether or abuse the system hints, but when they do use help, learning processes and outcomes may

be substantially improved. Another series of studies revealed that help-seeking errors are associated with poor learning (Baker et al. 2004; Alevén et al. 2006; Roll et al. 2014). These findings suggest that looking for right level of help in the right time will result in supporting learning. However, there are also various reasons that learners may not ask for help (such as fear that they will receive less credit for a successful outcome or being viewed as incompetent etc.)

2.2.2 *Open student modeling*

In open student modeling (OSM) approach, it offers a group of techniques that makes traditionally hidden student models available to the learner for exploration and possible editing. Representations of the student models vary, from displaying high-level summaries (such as skill meters) to charting out complex concept maps or networks. A spectrum of OSM benefits have been reported, such as increasing the learner's awareness of their own developing knowledge and difficulties in the learning process; as well as student engagement, motivation, and knowledge reflection (Bull et al. 2004; Hsiao et al. 2013; Mitrovic and Martin 2007; Zapata-Rivera and Greer 2004).

2.2.3 *Concept mapping*

Concept Mapping is an approach describes a representation of idea interpretation or perceived reality into a concepts and relationships link-node spatial arrangement. It is originated from science education to help students visualize their thinking structure and externalize knowledge (Novak 1990). The fundamental assumption of concept mapping is based on Ausubel's assimilation process learning theory (Ausubel 1968), suggesting that learning happens when the learner reconstructs or rearranges information in reducing the gap between a desired state and their own view of self. Over decades of development, numerous studies have reported positive outcomes that concept mapping facilitates meta-cognitive monitoring and reflection (Chang et al. 2001; Hwang et al. 2013; Novak 2002; Sanders and Stappers 2008). We aims to extend the lessons learned from concept mapping to address the challenge in articulating the dynamic state of novice-programming learners' knowledge in seeking information.

2.3 Modeling learning from discussion forums

Over the decades, discourse analysis on discussion forums has been carried out through various formats, network analyses, topical analyses, interactive explorers, knowledge extraction, etc. (Dave et al. 2014; Vassileva and Gutwin 2008). Due to calculation complexities (since linguistic features rely on computer processing power), most of these in-depth analyses were performed offline (Wen et al. 2014). As a result, the lesson learned could only be applied in the next iteration of system development. Recently, however, we have begun to see some studies that focus on dynamic support for users (Hoque et al. 2014). Yet, there has been no conclusive or comprehensive technological support, nor systematic studies to date on large-scale discussion forums that associate with students' learning. 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 et al.

(2013) 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 (2010) investigated online tutoring forums for homework help, making observations on the participation patterns and the pedagogical quality of the content; Hanrahan et al. (2012) and Posnett et al. (2012) studied expertise modeling in a similar sort of discussion environment; Goda and Mine (2011) quantify online forum comments by time series (previous, current and next) to infer the corresponding learning behaviors.

In this study, we focus on the StackOverflow (SO) discussion forum, which is a large-scale Q&A platform for programming learners to communicate. The Q&A posts on SO is taken as the resource of problem solving in our study, and the students' information seeking behaviors on the SO embed search engine is studied.

3 The first study: exploring and modeling programming information seeking

The goal of this study is to explore and to model students' programming information seeking behaviors. We investigate the behavioral differences between novices and advanced students and based on their sequential activity patterns, hidden Markov models are constructed to illustrate students' searching and reading processes. Finally, we analyze the content that students have read and examined, and discuss the association between searching & reading activities and their learning performances.

3.1 Data collection

We developed a Chrome browser plugin to support students' query behavior on SO site by entering their query to SO search engine. In SO search engine, after entering a query, the relevant posts found on SO is listed on the result page. For each post, the title, content snapshot, topic tags, the number of votes and answers are displayed. The performance of SO search engine is analyzed in the following subsections.

The browser plugin was offered to an Object-Oriented Programming class in the 2015 Fall semester at Arizona State University. Students were encouraged to install the browser plugin and use it to look for programming-related information throughout the entire semester. The plugin essentially collects students' queries and browsing activities (i.e. click, scroll, highlight) on SO. Additionally, all the activities are time-stamped and logged. The students are aware that their operations will be recorded for further study. Before the study, the students were also given a pre-test to examine their background knowledge about programming. According to the pre-test result, the students were split into novice group and advanced group by the median score, so the ratio of novices and advanced students is 1:1. The grouping aimed to study the behavior difference between novices and advanced students, while there were no difference in their class educating.

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 min. All the students' searching and reading behaviors on StackOverflow were recorded.

3.2 Data description

At last, 71 out of 86 students installed the plugin in the programming class, 640 queries and 423,942 operations were collected. The students' assignment and exam scores were also included as part of the data collection, which is used to measure the learning effect and match to their information seeking behavior.

3.2.1 Query data

For these 71 students installed the plugin, 55 students searched query on it. The average query number per student is 9.55 (max 56, min 1, median 8), and the average number per student of operations is 7179 (min 1, median 2917, max 140,300). 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 separated by whether the provider is novice or advanced student. The statistics is as follows in Fig. 1. As shown, the novices provided more query in average, but the length of each query is shorter, which indicated a lower quality according to Belkin's research (Belkin et al. 2003).

3.2.2 Overall behavioral patterns based on programming information seeking activities

We model search sessions into two phases: the searching phase and reading phase. In searching phase, student enter a query and browse the list of results, while in reading phase, the student click on one result and browse the content of linked web page. There could be back and forward between these two phases because after browsing a page, a student could go back to the result list and look for a better material.

There are 466,659 operations logged including scroll_up, scroll_down, click and select for both searching and reading phases. Figure 2 demonstrates the operations distribution at each phase. We found that for both groups of students, novices and advanced students, generated the majority of the operations in reading and in scrolling down. 19.3% operations are scrolling up in the searching phase in general. 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. This finding was supported by the previous user modeling studies (Cole et al. 2011). On the other hand, only 3.2% operations are clicks in searching phase, which indicated the challenge of trying to identify a relevant item from the massive forum posts of a given query. There were a few possible reasons that users did not have more

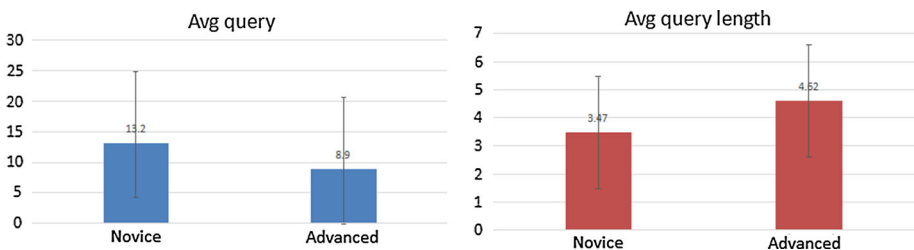


Fig. 1 Query information statistics

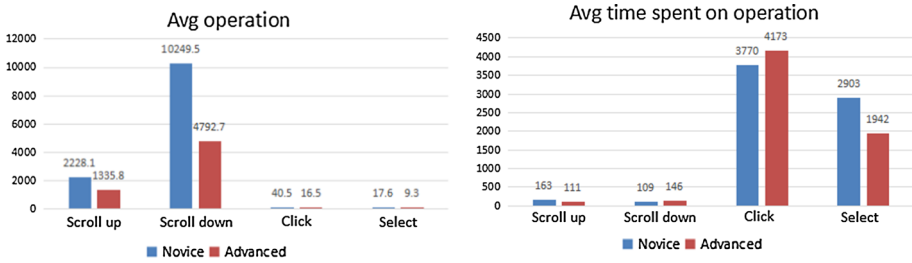


Fig. 2 Average number of operation and operation time (ms) distribution

clicks: (1) the queries were bad, so the results were not informative enough to drive users to read further; (2) the queries were good, but the users did not know how to judge whether the results were relevant or not. 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. However in reality, users need to scroll down when they do not feel satisfied with the results provided in the first place, and this unsatisfying ratio is reflected by the scrolling back and forward operation percentage.

This observation is also supported by the average time cost before each operation (Fig. 2). When browsing search results, users appear to spend more time (37.8%) before clicking or selecting, while they will be 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 spend more time on reading the content compare to advanced students. This finding is also supported by previous studies about expert bias (Kiseleva et al. 2015).

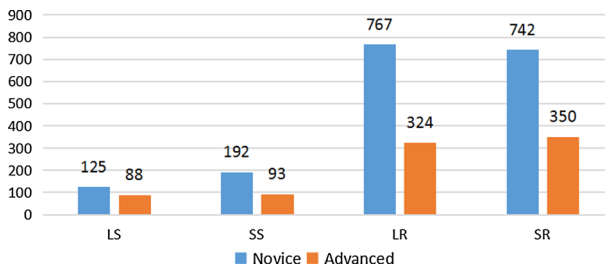
3.3 Dissecting programming information seeking sequential actions

In order to analyze students programming information seeking behavior on discussion forums, we categorize their actions into 6 categories based on Marchionini’s information seeking processes (Marchionini 2006): formulate queries, query refinement, results examination, and reading. We further split (by median) search and read phases into large-search (LS), small-search (SS), large-read (LR), small-read (SR) according to the amount of operations made on each single page, as a result, the ratio of LS and SS is 1:1, and the ratio of LR and SR is also 1:1. 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 Fig. 3. The reading actions are more than searching action, because all reading action on StackOverflow was captured in the data, including when a student used other search engine such as Google to find the thread page. Another fact is that novices searched and read more than advanced students. This is straightforward because novices would have to face more problems in learning, which motivated them to search and read more to solve the problems. However, whether more reading could lead to more learning is not determined, their behavior pattern also matters.

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

**Fig. 3** Average number of operation and operation time (ms) distribution

3.4 Model programming information seeking process with HMM

3.4.1 Model setup

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 (Han et al. 2013), survey design (Hsiao et al. 2014) and student learning process (Piech et al. 2012). 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 (Baum et al. 1970) can be used to learn the emission and transition probabilities.

3.4.2 Apply HMM on information seeking processes

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 complex model with large number of states will help to increase the sequence likelihood because there are more parameters that can be used to describe the model more precisely. But it has a high risk to cause over-fitting. A simple model is less likely to over-fit on the given data set, but it may not be able to uncover the natural feature of data sets. It is still an open issue for determining the number of hidden states, which is a model selection problem in parameter learning of hidden Markov model. In model selection, the information criterion such as the Akaike information criterion (AIC) or its variants Bayesian information criterion (BIC) (Baum et al. 1970) 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 (Fig. 4).

According to AIC, hidden state transition analysis was applied on Novice and Advanced group, the hidden state transition diagrams is shown in Fig. 5 for better navigation and comparisons.

3.5 Evaluation of analysis results

The goal of analyzing the data was to highlight the difference between novices and advanced students including querying customs, browsing behavior patterns, and topic of content browsed. Clustering, sequence mining, and topic detection techniques were applied to achieve the goals.

3.5.1 Novices lack the ability to examine query results

As show in Fig. 5, advanced students consistently perform query refinements (3:1 ratio) before they examine the results (HS3 HS1). Novices behave differently. Part of them follows the similar pattern as advanced students do, tuning the queries before examine the results (HS4 HS1). However, when these novices refine queries, there are no consecutive

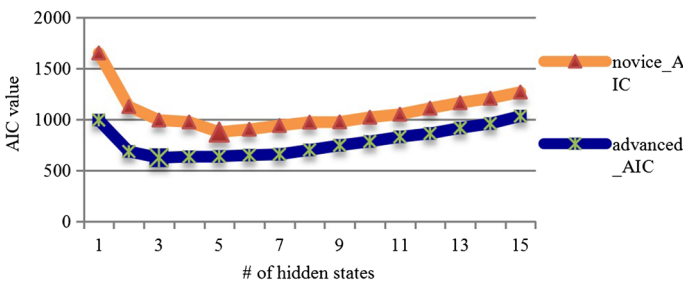


Fig. 4 Choosing number of hidden state using AIC

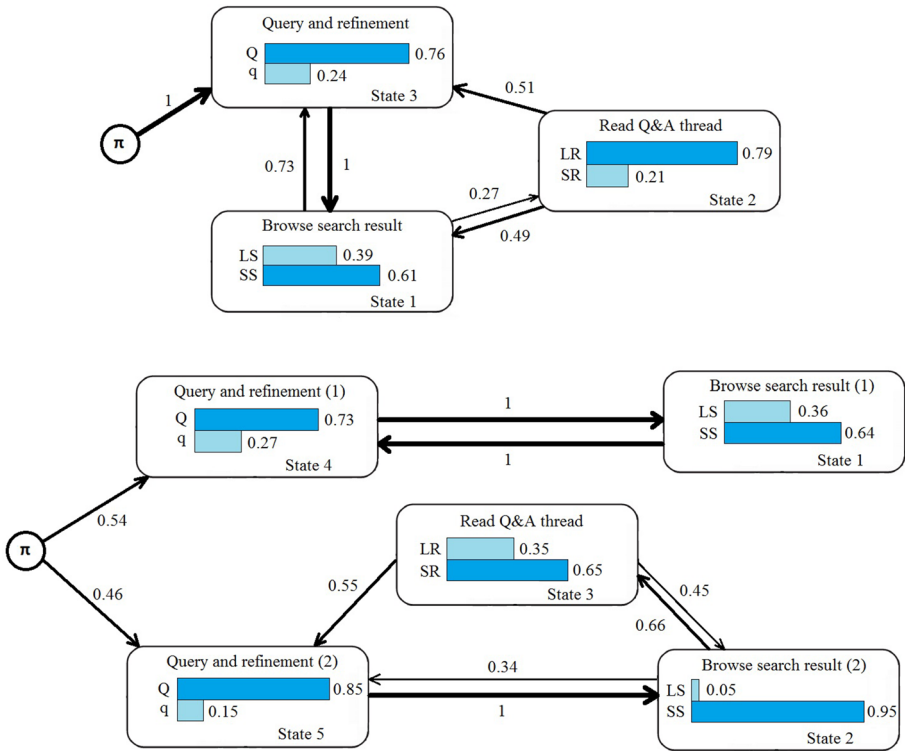


Fig. 5 Advanced (*up*) and Novice (*bottom*) students information seeking transition probability diagrams

actions followed in the next step (Fig. 5 bottom), which indicated that they did not go to any reading page. On the other hand, when novices do minimum query refinements (HS5 HS2), they do manage to proceed to next step, which is the reading phase (HS5 HS2 HS3). It suggests 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 are small searches, which means that novices tend not to scrutinize the search results. They only examine the results minimally. Even move on to read forum posts (HS5 HS2 HS3), they can be reading whatever the discussion forum has recommended (i.e. top returned items).

In fact, Fig. 6 shows the total amount of time that each student spent on searching or reading pages. It is surprising to see that novices students spent more than 130 min on just reading, while advanced students spent about 40 min. 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. However, does more reading of discussions mean more learning? What exactly do students read on the discussion forum? We further look into students' reading behavior and reading content in the following section. Despite the reading quality, based on novices' behaviors can also suggest the hidden danger of online large-scale

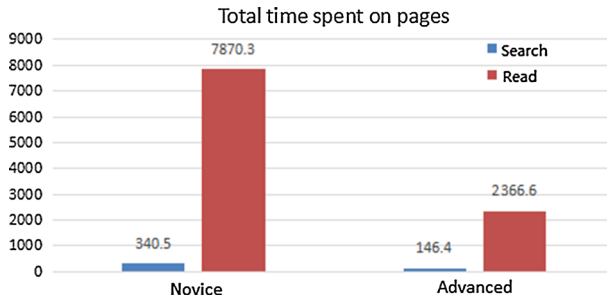


Fig. 6 Total time spent on searching and reading average per student

discussion forums, where the existing filtering mechanisms (such as badges, acceptance, votes) may not be enough, especially for novices.

3.5.2 Novices have difficulty in forming query

When students eventually land on forum posts pages and read, we found that advanced students commit to careful reading, as oppose to novices' careless reading (Advanced HS2: 0.79 LR; Novice HS3: 0.65 SR). In fact, we found that students did spend time on reading the pages, and novices cost more time in small reading than advanced students, while in large reading advanced students spent slightly more time, but no significant difference between groups. The modeling results along with time spent statistics reveal that given not likely that novices perform thorough search results filtering, but once they did, they would spend time to read. Thus, it led us to examine their initial state, queries. Do novices and advanced students have query pattern differences? For instance, are advanced students better in forming queries or refining queries? Are advanced students better at filtering search results?

We calculated cosine similarity between each pair of adjacent queries for each student. We found that the daily search average similarity was not high (0.35 ± 0.34), and most adjacent queries did not share any common words (similarity is 0), which means the students were searching about different topics. However, in lab class the students search much more similar queries, which means they had to refine the query for more times to achieve satisfying search result. This fact is reasonable because in lab class, the students were given a complex task and required to finish in limited time period, so they are more eager to find suitable resource to reuse. The similarity distributions of both daily search and lab class search are as shown in Fig. 7.

3.5.3 Novices refine query with obstacles under pressure

When we compare students' query similarity patterns by novices and advanced students (Fig. 7), we found that novices are not significantly different from advanced students in daily searches. In another word, novices and advanced student behave similarly in refining their queries ($p = 0.2377$). However in the lab class, when time is limited to finish a task, these two groups appeared different query refinement tendency. Novices searched more similar queries in the lab class. It suggested that novices may be under limited time pressure, therefore, their ability to refine queries is affected and leads to more steps in the refinements to find a satisfying result, or even quit the search without any reading.

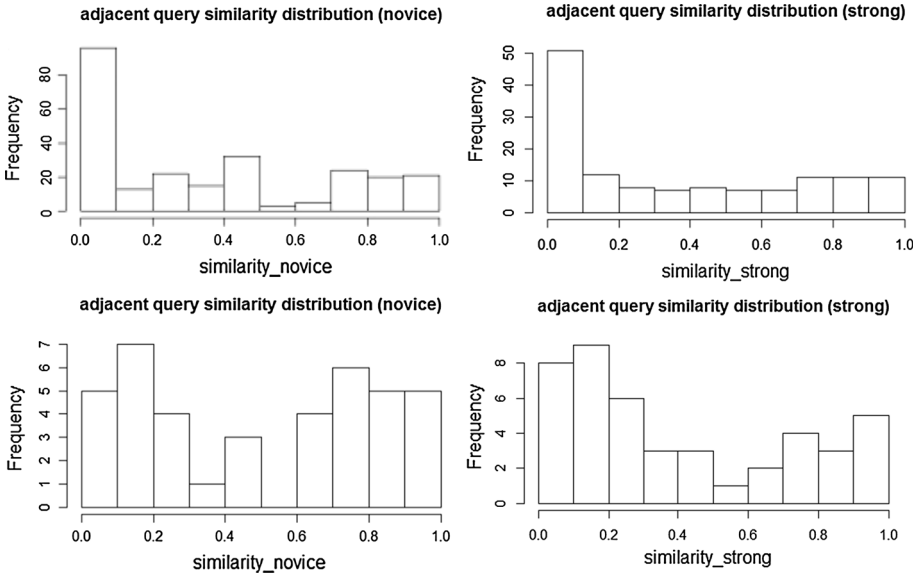


Fig. 7 Frequency distribution of adjacent query similarity for daily search and lab class search

3.5.4 Both groups read posts based on course schedule topics

We crawled all the posts that students read from SO, and performed text mining. We modeled the text to summarize topic words using MALLET² LDA toolkit with default $\alpha = 30/N$, $\beta = 0.01$, $itr = 1000$. We found students were reading the topics from discussion forums according to the course schedule, from week 1 *JavaBasis* 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 (Fig. 8). 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 topic: *Recursive* (Table 2 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 weeks 4 and 9, advanced students were found to be less focused in terms of reading more diverse topics was due to those 2 weeks were exam periods.

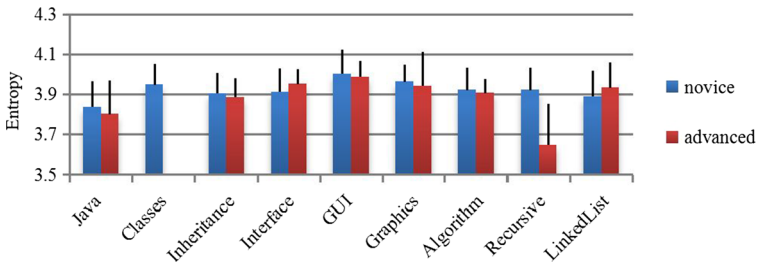


Fig. 8 Weekly readings' topic focus by novices and advanced students

Table 2 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, mathematic, call, line, learn, number, work, value, function, undefined}

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

Therefore, it is understandable that students might read a wider range of topics that were covered over exam periods.

3.5.5 Advanced students read more in-depth and technical content

In the controlled lab session, advanced students were curious not only about the implementation, but also deeper knowledge. Three advanced students read the same post on the discussing differences between merge sort and quick sort, which was not closely related to their task. On the other hand, advanced students also read detailed technique posts rather than general method discussion threads. E.g. “scannervsbufferedReader” is detected as a hot topic among advanced students, it is a detailed topic related to the merge sort implementation, while hot topic detected for novices are generic titled posts, such as “recursivemergesort” or “mergesortjava”. This observation indicated that advanced students had already had the general implementation in mind, so they constructed sub-tasks and proceeded to find relevant detailed technical information,, while novices were still looking for general implement idea.

3.5.6 The more read, the higher score in exam

Based on the percentage of large read rate in reading pages, we found that the more students spending time in reading on SO, 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 9 shows the connection between large read rate and final exam score.

4 The second study: Personalized Information Seeking Assistant (PiSA), a tool facilitates programming information seeking

Based on what we have learned from students’ programming information seeking behaviors (Sect. 3), we have identified the learning challenges for programming novices. In this section, we propose a personalized information seeking tool to assist programming novices seek for related information. The system is named Personalized Information Seeking Assistant (PiSA), which works as a search engine where programming learners can enter a query and receive adaptive help to find programming related discussions from SO. The purpose of designing PiSA system is attempting to remove the gap between programming novices and the professional community, to help novices seek information

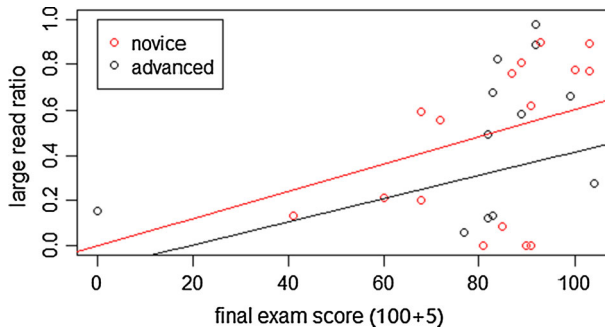


Fig. 9 Final score versus Large read rate

easily and to maximize learning opportunity during the searching process. The following sections describes the design rationales for the proposed system and presents the evaluation results.

4.1 PiSA design rationale

4.1.1 Search result summary

Since there are clues showing that the novices have difficulty in forming and evaluating query when seeking for problem solution (Sect. 3.5), we purposefully design a feature of summarizing search results to facilitate query preview. The summary is visualized as bubble chart, it includes two views: tag view and word view. Each bubble in the chart represents a tag/word appear in the results, in which the size of bubble indicates the total frequency of the tag/word appears.

The effect of search result summary helps information seeker to get a general cognition of the results before they browse each of them, and quickly figure out whether the results are closely relevant to their original seeking purpose. Since novices have difficulty to quickly understand materials (Lu and Hsiao 2016), it is extremely hard for them to realize that the results are irrelevant when their query is improper.

4.1.2 Browse history summary

Besides summarizing the search result, another necessary assistance to the novices is to give term suggestions when generating queries. In PiSA, this assistance is provided by summarizing the user's personal browse history. This history is summarized as a serial of bubble charts, it represents the tags/words of Q&A threads browsed in each week. Similar to result summary, the size of bubble indicates the frequency of tag/word. In this way, learners could easily trace their personal learning path, and recall what was browsed. Moreover, by listing the browsed tags and words, learners could see potential good terms for their first query, and the search result summary could continue to provide term suggestion to refine the query.

4.1.3 Social navigation support

In order to motivate learners to search and read, social feature is involved. Freyne et al. (2007) studied about the community wisdom in social search and social navigation, which highlighted the value of users search and browse history as a social support for information seeking. In PiSA, we involved the social feature by suggesting the browse frequency of the whole community for each tag/word in summaries.

In both history and search result summary, the color of bubble indicates the “popularity” of the tag/word. When a tag/word is frequently browsed by many learners in the same community, its bubble will be colored darker. In this way, learners could easily find the hot topics for each week in their own browse history, and realize their query is good when the result summary is generally darker. Moreover, a learner could easily identify the concepts they overlooked when the bubble appears to be dark but small, which means it is browsed frequently by the community but he/she missed it.

4.2 Query refinement

With the help of search result summary, learners could realize when it is necessary to refine the query. Furthermore, Silverstein’s study (Silverstein et al. 1999) showed that novices could learn terms from previous search result, and use these new terms directly in the refined query even though they did not understand the term clearly. As discussed in Silverstein’s study (Silverstein et al. 1999), the behavior of using new terms itself is a learning progress, learners benefit from even realizing new concepts. In this degree, PiSA provides the terms to learners directly in search result summary before they read, which makes it faster to refine the query with new terms.

In PiSA, users could add terms in the summary to their new query by simply clicking on the bubble. The purpose of this design is to encourage learners to use new terms, and attract them to pay more attention to the bubble charts when they have problem in understanding content in search result.

If searching is not enough to learn about the concept in bubbles, a more straight forward way is to view it on Wikipedia. PiSA provides such connection to encourage learning, user could view the instruction of terms in bubble by simply clicking the text in bubble.

Moreover, another feature provided about query refinement is to “exclude” terms. Novices have a lack of background knowledge and concepts, so misuse of concepts happens a lot, which leads to the problem that irrelevant results conceal the wanted one. PiSA solve this problem by excluding specific term, users could easily do it by right clicking the bubble.

4.3 Document & API assistant

One of the most shortage according to early user study in PiSA is the volume of document set. Since PiSA only searches in SO Q&A threads, it mostly solves problems about errors and coding, but helps little in concept interpreting.

In order to improve the quality, document and API assistant is added into the search feature. When keywords about programming language such as “java”, “html”, or “php” are detected, PiSA will provide the link to the standard document websites querying their problem, so users could choose to browse the instructions related to their query before reading the Q&A threads.

5 PiSA evaluation

In this section, we evaluate PiSA system by a 1-month experiment on the students of a programming course. The expectations are as follows:

- With the assistance from PiSA, more students could find the information they were seeking;
- Students spend less time and less operations to find the information they were seeking;
- Students spend more time on finely reading the material they found from PiSA since it provides more useful information.

In the evaluation section, the embed search engine in SO is compared with PiSA as a baseline. The reason is that PiSA accesses documents from SO API, so the search and rank result of PiSA is exactly the same as SOs, the only difference between PiSA and SO search engine is the interface and visual features of PiSA (Fig. 10).

5.1 Data collection

At the moment of writing, we have collected 1-month worth of search behavioral data from a graduate level programming course, offered by Arizona State University. Students were asked to register account and encouraged to use PiSA when encounter programming problems solving, they were also aware that their exam and assignment scores are anonymously used in further study. The purpose to establish the 1-month study instead of a short lab experiment is to collect learners' natural behaviors on searching solutions from online discussion forums. There were total 34 students recorded of using PiSA, including 73 queries and 1392 operations (click, scroll, select text).

5.2 Classroom study

During the month of experiment, in average each student provided 3.48 ± 2.89 queries, applied 9.09 ± 26.79 operations on the main page if PiSA (Fig. 11), and 31.85 ± 69.22 operations on the search result page (Fig. 12).

This ratio indicates that students browse more on search result compare to their history. It is reasonable since PiSA is still in cold start phase, and students had little history to browse at the beginning. In order to guarantee the analysis is reliable, the 50% students

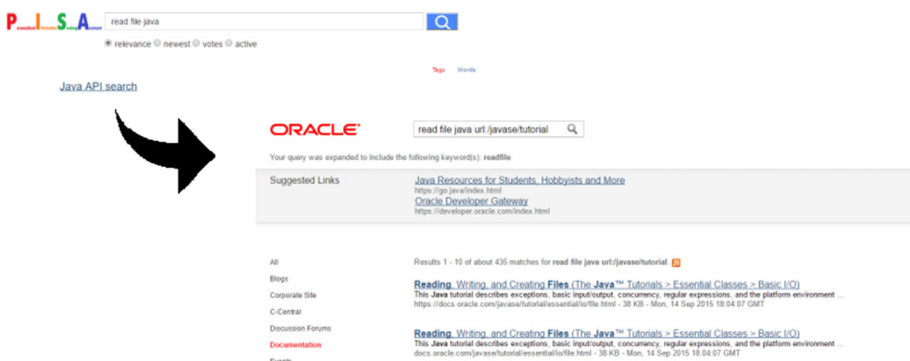


Fig. 10 Document & API assistant in PiSA

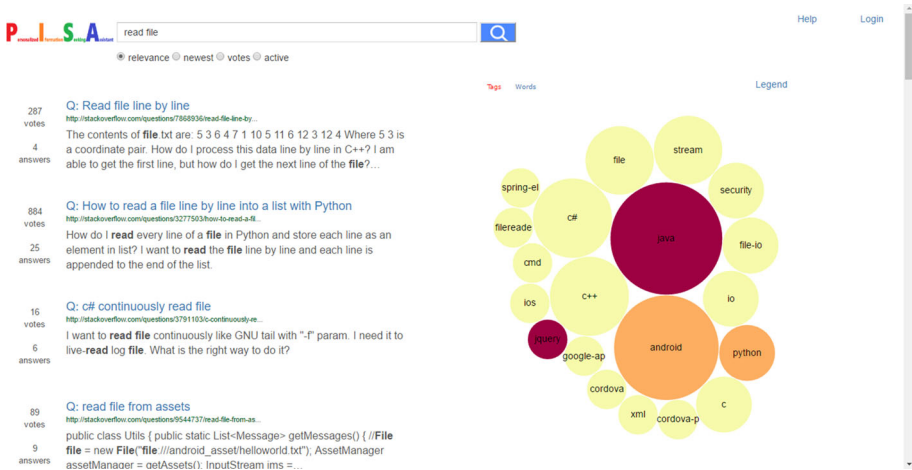


Fig. 12 PiSA Search result and summary effect

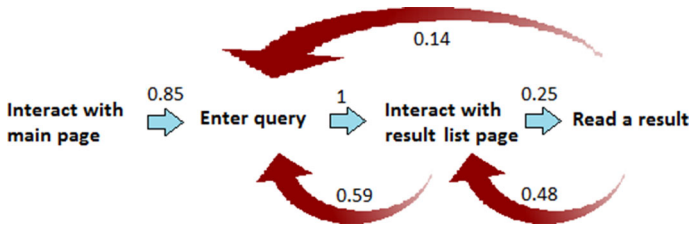


Fig. 13 User behavior clustering

5.3.2 Students’ operation pattern clusters

The general sequence pattern study did not consider the variance of individual students, but in reality students’ behavior are vary in information seeking (Mitrovic and Martin 2007), so a study on individual patterns is established based on their operation sequence. This study considers the page of each operation as the status, and analysis one-step transformation. So each student has a vector representing the ratios of transfer from one page to other, or stay on the same page.

Based on the vector space of students, a k-mean clustering was applied to mine the vary patterns of student operation sequences. The result shows the students are best separated into 3 clusters, and the most significant variable is the ratio of transfer between “main page—main page”, “result list page—result list page”, and “result list page—SO page”. The cluster result is shown in Fig. 14.

In cluster 1 ($N = 4$), the students are balanced in operating on main page and result list page, and they have the highest ratio of transfer to SO page. This fact indicates that students in cluster 1, who also pay attention to their browse history, actually have a better chance to find materials to read.

In cluster 2 ($N = 6$), the students mostly operation on result list page, but have little ratio of transfer to SO pages. It means students in this cluster are stuck at the result list, they have problem in choosing a relevant result to read, or more commonly, their query

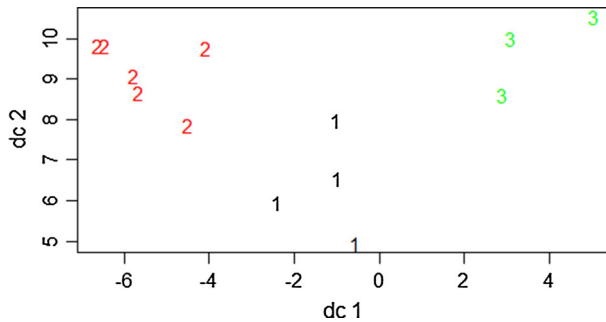


Fig. 14 User clustering by sequence character

quality is not high enough to find a relevant page. Another fact about cluster 2 is that their ratio of refining query (enter query after viewing result list) is not high, which means they still have problem in refining query.

In cluster 3 ($N = 3$), students mostly stay in the main page, in this case they did not do much searching, they are relatively the inactive users in the system.

6 Conclusions and future work

6.1 Result discussions

In this work, we investigate the issues of how do students learn programming during search. Specifically, we investigate in how do novice students look for programming-related information from large-scaled discussion forum and how do they learn from searching. We conduct two classroom studies to explore programming information seeking strategies and design PiSA to facilitate programming novices look for programming related information.

In the first study, we identified there are distinct behavior differences between programming novices and advanced learners when seeking for information, which can be classified as a support of expertise bias (Kiseleva et al. 2015). We model these learners' query formulation, refinement, results examination, and reading processes with hidden Markov model. We conduct sequential pattern mining with hidden Markov model. 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. Students progressively read the discussion with topical posts according to the course schedule. The study also uncovers that the programming novices usually spend more time in browsing search result and reading, which is the consequence of the lack of pre-knowledge. However, 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.

In the second study, we designed PiSA system aiming to assist programming novices to learn while seeking programming-related information. PiSA utilizes multiple visual elements to help summarize search results, evaluate query, and provide term suggestions in

query. PiSA also applies social navigation support to integrate the user's own browsing history to assist query discovery and expansion. Behavioral pattern mining results indicate that PiSA help in query refinement. A further clustering analysis also reveals that students who pay attention to their browsing history lead to further reading events, which subsequently resulting in potential learning activities (per our findings in Sect. 3).

6.2 Limitations and future work

In order to maximize the value of PiSA in programming education, here are a number of future directions for future research. (1) We plan to conduct more comprehensive user evaluations on PiSA, including measuring longer term of behavior monitoring. Currently, the data collection was limited within 1-month period, which may not be representative enough to capture students' learning process. (2) The document set could be expended to other educational resources (i.e. electronic textbooks) other than merely online discussion threads. For instance, we can incorporate Google's search results into our search results pool. (3) Currently, students' queries and query suggestions are adapted to theirs and peers' histories in adaptive navigational form. Based on what we have learnt from students' searching behaviors, we can improve PiSA's personalization by providing more proactive personalization, such as query recommendations or reading recommendation.

References

- Ageev, M., Guo, Q., Lagun, D., & Agichtein, E. (2011). Find it if you can: A game for modeling different types of web search success using interaction data. In *SIGIR*.
- Aleven, V., McLaren, B., Roll, I., & Koedinger, K. (2006). Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education*, 16(2), 101–128.
- Aleven, V., Roll, I., McLaren, B. M., & Koedinger, K. R. (2016). Help helps, but only so much: Research on help seeking with intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 26(1), 205–223.
- Ausubel, D. (1968). *Educational psychology: A cognitive view*. New York: Holt, Rinehart & Winston.
- Baker, R. S., Corbett, A. T., & Koedinger, K. R. (2004). Detecting student misuse of intelligent tutoring systems. In *International conference on intelligent tutoring systems* (pp. 531–540). Springer, Berlin.
- 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.
- Beal, C., Mitra, S., & Cohen, P. (2007). Modeling learning patterns of students with a tutoring system using hidden Markov model. In R. Luckin et al. (Eds), *Proceedings of the 13th international conference on artificial intelligence in education (AIED)*, Marina del Rey.
- Belkin, N. J., Kelly, D., Kim, G., Kim, J. Y., Lee, H. J., Muresan, G., et al. (2003). Query length in interactive information retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 205–212). ACM.
- Bilal, D. (2002). Children's use of the Yahoo!igans! Web search engine. III. Cognitive and physical behaviors on fully self-generated search tasks. *Journal of the American Society for Information Science and Technology*, 53(13), 1170–1183. doi:10.1002/asi.10145.
- Boyer, K. E., Phillips, R., Ingram, A., Ha, E. Y., Wallis, M., Vouk, M., et al. (2011). Investigating the relationship between dialogue structure and tutoring effectiveness: A hidden Markov modeling approach. *International Journal of Artificial Intelligence in Education*, 21(1–2), 65–81.
- Brandt, J. (2010). *Example-centric programming: Integrating web search into the development process*. Ph.D., Stanford University.
- Bull, S. (2004). Supporting learning with open learner models. Paper presented at the 4th Hellenic conference on information and communication technologies in education, Athens.

- Bull, S., Ginon, B., Boscolo, C., & Johnson, M. (2016). Introduction of learning visualisations and metacognitive support in a persuadable open learner model. In *Proceedings of the sixth international conference on learning analytics & knowledge* (pp. 30–39). ACM.
- Chang, K. E., Sung, Y. T., & Chen, S. F. (2001). Learning through computerbased concept mapping with scaffolding aid. *Journal of Computer Assisted learning*, 17(1), 21–33.
- Chi, E., Pirolli, P., & Lam, S. (2007). Aspects of augmented social cognition: Social information foraging and social search. In D. Schuler (Ed.), *Online communities and social computing* (Vol. 4564, pp. 60–69). Berlin: Springer.
- Choo, C. W., Detlor, B., & Turnbull, D. (2000). Information seeking on the web: An integrated model of browsing and searching. *First Monday*, 5(2). doi:10.5210/fm.v5i2.729
- Cole, M. J., Zhang, X., Liu, C., Belkin, N. J., & Gwizdka, J. (2011). Knowledge effects on document selection in search results pages. In *SIGIR* (pp. 1219–1220).
- Cutrell, E., & Guan, Z. (2007). What are you looking for?: An eye-tracking study of information usage in web search. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 407–416). ACM.
- Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in GitHub: Transparency and collaboration in an open software repository. Paper presented at the proceedings of the ACM 2012 conference on computer supported cooperative work, Seattle, Washington.
- Dave, K., Wattenberg, M., & Muller, M. (2004). 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* (pp. 232–241). ACM.
- Evans, B. M., & Chi, E. H. (2008). Towards a model of understanding social search. Paper presented at the proceedings of the 2008 ACM conference on computer supported cooperative work, San Diego, CA.
- Freyne, J., Farzan, R., Brusilovsky, P., Smyth, B., & Coyle, M. (2007). Collecting community wisdom: Integrating social search & social navigation. In *Proceedings of the 12th international conference on intelligent user interfaces* (pp. 52–61). ACM.
- Goda, K., & Mine, T. (2011). Analysis of students learning activities through quantifying time-series comments. In *International conference on knowledge-based and intelligent information and engineering systems* (pp. 154–164). Springer, Berlin.
- 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*, London.
- Han, S., Yue, Z., & He, D. (2013). Automatic detection of search tactic in individual information seeking: A hidden Markov model approach. *arXiv preprint arXiv:1304.1924*.
- Hanrahan, B. V., Convertino, G., & Nelson, L. (2012). Modeling problem difficulty and expertise in stackoverflow. In *Proceedings of the ACM 2012 conference on computer supported cooperative work companion* (pp. 91–94). ACM.
- Hearst, M. (2009). *Search user interfaces*. Cambridge: Cambridge University Press.
- Holmes, R., & Murphy, G. C. (2005). Using structural context to recommend source code examples. Paper presented at the proceedings of the 27th international conference on software engineering, St. Louis, MO.
- Hoque, E., Carenini, G., & Joty, S. (2014). *Interactive exploration of asynchronous conversations: Applying a user-centered approach to design a visual text analytic system* (p. 45). Sponsor: Idibon.
- Hsiao, I. H., Han, S., Malhotra, M., Chae, H. S., & Natriello, G. (2014). Survey sidekick: Structuring scientifically sound surveys. In *International conference on intelligent tutoring systems* (pp. 516–522). Springer.
- Hsiao, I.-H., Li, Q., & Lin, Y.-L. (2008). Educational social linking in example authoring. Paper presented at the proceedings of the nineteenth ACM conference on hypertext and hypermedia, Pittsburgh, PA.
- Hsiao, I.-H., Bakalov, F., Brusilovsky, P., & Knig-Ries, B. (2013). Progressor: Social navigation support through open social student modeling. *New Review of Hypermedia and Multimedia*, 19(2), 112–131.
- Hwang, G. J., Yang, L. H., & Wang, S. Y. (2013). A concept map-embedded educational computer game for improving students' learning performance in natural science courses. *Computers & Education*, 69, 121–130.
- Jeong, H., Gupta, A., Roscoe, R., Wagster, J., Biswas, G., & Schwartz, D. (2008). Using hidden Markov models to characterize student behaviors in learning-by-teaching environments. In *International conference on intelligent tutoring systems* (pp. 614–625). Springer, Berlin.
- Joachims, T., Granka, L., Pan, B., Hembrooke, H., & Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 154–161). ACM.

- Kiseleva, J., Garca, A. M., Kamps, J., & Spirin, N. (2015). The impact of technical domain expertise on search behavior and task outcome. *arXiv preprint arXiv:1512.07051*.
- Ko, A. J., Myers, B. A., Coblenz, M. J., & Aung, H. H. (2006). An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks. *IEEE Transactions on Software Engineering*, 32(12), 971–987. doi:10.1109/TSE.2006.116.
- Krikelas, J. (1983). Information-seeking behavior: Patterns and concepts. *Drexel Library Quarterly*, 19(2), 5–20.
- Lorigo, L., Haridasan, M., Brynjarsdóttir, H., Xia, L., Joachims, T., Gay, G., et al. (2008). Eye tracking and online search: Lessons learned and challenges ahead. *Journal of the Association for Information Science and Technology*, 59(7), 1041–1052.
- Lu, Y., & Hsiao, I-H. (2016). Seeking programming-related information from large scaled discussion forums, help or harm? In *The 9th international conference on educational data mining, Raleigh, NC*.
- Marchionini, G. (2006). Exploratory search: From finding to understanding. *Communications of the ACM*, 49(4), 41–46.
- Mitrovic, A., & Martin, B. (2007). Evaluating the effect of open student models on self-assessment. *International Journal of Artificial Intelligence in Education*, 17(2), 121–144.
- Mujumdar, D., Kallenbach, M., Liu, B., & Hartmann, B. (2011). Crowdsourcing suggestions to programming problems for dynamic web development languages. In *CHI'11 extended abstracts on human factors in computing systems* (pp. 1525–1530). ACM.
- Nasehi, S. M., Sillito, J., Maurer, F., & Burns, C. (2012). What makes a good code example?: A study of programming Q&A in StackOverflow. Paper presented at 2012 28th IEEE international conference on the software maintenance (ICSM).
- Novak, J. D. (1990). Concept mapping: A useful tool for science education. *Journal of research in science teaching*, 27(10), 937–949.
- Novak, R. W. (2002). Do proficiency testing participants learn from their mistakes? Experience from the EXCEL throat culture module. *Archives of Pathology & Laboratory Medicine*, 126(2), 147–149.
- Piech, C., Sahami, M., Koller, D., Cooper, S., & Blikstein, P. (2012). Modeling how students learn to program. In *Proceedings of the 43rd ACM technical symposium on computer science education* (pp. 153–160). ACM.
- Pirolli, P., & Card, S. (1999). Information foraging. *Psychological Review*, 106(4), 643.
- Posnett, D., Warburg, E., Devanbu, P., & Filkov, V. (2012). Mining stack exchange: Expertise is evident from initial contributions. In *2012 International conference on social informatics (SocialInformatics)* (pp. 199–204). IEEE.
- Puustinen, M., & Rouet, J. F. (2009). Learning with new technologies: Help seeking and information searching revisited. *Computers & Education*, 53(4), 1014–1019.
- Reinecke, K., Yeh, T., Miratrix, L., Mardiko, R., Zhao, Y., Liu, J., et al. (2013). Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 2049–2058). ACM.
- Roll, I., Baker, R. S. D., Aleven, V., & Koedinger, K. R. (2014). On the benefits of seeking (and avoiding) help in online problem-solving environments. *Journal of the Learning Sciences*, 23(4), 537–560.
- Sanders, E. B. N., & Stappers, P. J. (2008). Co-creation and the new landscapes of design. *Co-design*, 4(1), 5–18.
- Silverstein, C., Marais, H., Henzinger, M., & Moricz, M. (1999). Analysis of a very large web search engine query log. In *ACM SIGIR forum* (Vol. 33, No. 1, pp. 6–12). ACM.
- Singer, J., Elves, R., & Storey, M. (2005). NavTracks: Supporting navigation in software. Paper presented at proceedings of the 13th international workshop on the program comprehension, 2005. IWPC 2005.
- Stylos, J., & Myers, B. A. (2006, 4–8 Sept. 2006). Mica: A web-search tool for finding API components and examples. Paper presented at the IEEE symposium on the visual languages and human-centric computing, 2006. VL/HCC 2006.
- Teevan, J., Alvarado, C., Ackerman, M. S., & Karger, D. R. (2004). The perfect search engine is not enough: A study of orienteering behavior in directed search. Paper presented at the Proceedings of the SIGCHI conference on human factors in computing systems.
- Treude, C., Barzilay, O., & Storey, M. (2011). How do programmers ask and answer questions on the web?: NIER track. Paper presented at the 2011 33rd international conference on software engineering (ICSE).
- van de Sande, C. (2010). Free, open, online, mathematics help forums: The good, the bad, and the ugly. In *Proceedings of the 9th international conference of the learning sciences* (Vol. 1, pp. 643–650). International Society of the Learning Sciences.

- Vasilescu, B., Serebrenik, A., Devanbu, P., & Filkov, V. (2014). How social Q&A sites are changing 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.
- Vassileva, J., & Gutwin, C. (2008). Exploring blog archives with interactive visualization. In *Proceedings of the working conference on advanced visual interfaces* (pp. 39–46). ACM.
- Wen, M., Yang, D., & Rose, C. (2014). Sentiment analysis in MOOC discussion forums: What does it tell us? In *Educational data mining 2014*.
- White, R. W., & Drucker, S. M. (2007). Investigating behavioral variability in web search. Paper presented at the Proceedings of the 16th international conference on World Wide Web.
- Wildemuth, B. M., & Freund, L. (2012). Assigning search tasks designed to elicit exploratory search behaviors. Paper presented at the Proceedings of the symposium on human-computer interaction and information retrieval, Cambridge.
- Wilson, M. L., & Schraefel, M. C. (2008). Improving exploratory search interfaces: Adding value or information overload? Paper presented at the second workshop on human-computer interaction and information retrieval, Redmond, WA.
- 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.
- Zapata-Rivera, J. D., & Greer, J. E. (2004). Interacting with inspectable bayesian student models. *International Journal of Artificial Intelligence in Education*, *14*(2), 127–163.