INSIGHT: a Semantic Visual Analytics for Programming Discussion Forums

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

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

ABSTRACT
This paper presents INSIGHT, a visual analytics web application, designed to induce & inspire programming language learning from discussion forums. The visual analytics, extracts and displays semantic content from 'Stack Exchange' in a form of bubble chart. The bubbles represent summarized semantic concepts from the forum posts and outlines the concept specificity of each individual post. The discussion forum content are modeled as concepts based on an innovative Topic Facet Modeling algorithm (a probabilistic topic model that assumes all words in single sentence are generated from one topic facet), and aimed to provide better understanding and solicitation of the increasing large volume of discussion content. We hypothesize that by navigating and interacting (browsing, sorting, searching etc.) with the Facets, will enhance learning. A comprehensive system design rationales and preliminary qualitative study are reported in this paper.

Categories and Subject Descriptors
H.4 [**Information Systems Applications**]: Miscellaneous; K.3.2 [**Computer and Information Science Education**]: Computer Science Education

General Terms
Measurement, Design, Experimentation, Programming, web application.

Keywords
Learning Analytics, discourse analysis, visual analytics, programming, discussion forums, computer-supported collaborative learning.

1. INTRODUCTION
Learning programming involves a variety of complex cognitive activities, from conceptual knowledge construction to basic structural operations, program design, programming understanding, modifying, debugging, and documenting (Lye & Koh, 2014; Piech et al., 2012; Robins, Rountree, & Rountree, 2003). There have been major educational technology advances over the last two decades, centered on understanding the nature of programming skills explicitly using declarative aspects of programmer's knowledge (i.e. program comprehension and generation, required concepts & skills to program). For example: intelligent tutors, auto program feedback generation, collaborative programming support, personalized learning resources, etc. (Aleven, McLaren, Roll, & Koedinger, 2006; Anderson & Skwarecki, 1986; Atkinson & Renkl, 2007; Barnes & Stamper, 2008; Boyer et al., 2011; Hsiao, Sosnovsky, & Brusilovsky, 2010; Lye & Koh, 2014; Piech, Sahami, Koller, Cooper, & Blikstein, 2012; VanDeGrift, 2004) The technology support has evolved from classrooms to online, declarative to exploratory, and individual to social. 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 examples). The internalization process from forming a question to reaching out to APIs or textbooks is usually not captured in learning programming. From a constructivism point of view, the action of articulating a problem and initiating search or referencing can be a valuable learning activity. There are numerous tools that have been built to make completing programming tasks easier, such as Mica (Stylos & Myers, 2006) (there are more cases reviewed in the literature review section), but less is focused on amplifying learning opportunities.

In the easily accessible Internet era, search engines, index and make the excessive amount of programming problems and solutions available. Because programming problems are usually more complex than a simple sequence of query keywords, dedicated communities such as discussion forums and Q&A sites are the most popular alternatives for problems & solutions. The drastic shift in momentum of learning opportunities from APIs and textbooks to community help is not yet fully comprehended though. Besides, forums or discussion boards usually lack dynamic and extensive content analysis due to large and increasing content volume and high computational cost in discourse analyses. In this work, we aim to research a new technology to facilitate online learning from programming discussion forum. We apply Learning Analytics approach, which has demonstrated promising results in online learning (Siemens & Baker, 2012). However, the majority of learning analytics focuses on visual representations or the system's usefulness, the core should be focused on the visualization impact to improve learning or teaching

(Verbert, Duval, Klerkx, Govaerts, & Santos, 2013). In this work, we present a new visual analytics system that targets at providing better understanding and solicitation of the increasing large volume of discussion content.

In the rest of the paper, we summarize the related work in learning analytics and other intelligent visual support for programming language learning. We then describe briefly the methodology to extract forum content semantics. In section 4, we present the system design and rationales. A user study and preliminary results are presented in section 5 & 6. Finally, we summarize the work and discuss future work and limitation.

2. LITERATURE REVIEW

2.1 Learning Analytics

Signals project at Purdue University is one of the pioneering examples of the successful application of academic analytics that integrate predictive modeling and report significantly higher grades and retention rates than were observed in control groups (Arnold, 2010). Septris and SICKO project at Stanford School of Medicine utilizes educational simulation games to offer deeper insight into learner's competency and decision making to help prepare doctors well. The game analytics not only help instructors see what choice learners made but also what data was used to make those choices and when they decided to make those choices. (Jamie Tsui, James Lau, Lisa Shieh, 2014). The application has been well received by the learners and instructors with over 32000 usage, 16000 plays and 2500 completions.

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, Wattenberg, & Muller, 2004; Gretarsson et al., 2012; Indratmo, Vassileva, & Gutwin, 2008; Lee, Kim, Cho, & Woo, 2013; Wei et al., 2010). 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, (Sande, 2010) investigated online tutoring forums for homework help by making observations on the participation patterns and the pedagogical quality of the content. (Hanrahan, Convertino, & Nelson,2012; Posnett, Warburg, Devanbu, & Filkov, 2012) studied expertise modeling in such environment. Cohere (Shum,2008) investigates semantic connections by identifying the link types to associate negative, positive, neutral interactions among online discourses. (Wise, Zhao, & Hausknecht,2013) observed the listening behavior, which encapsulates different actions that learners take in relation to others posts (attending, reading etc.), to further describe the discussion engagement.

2.2 Intelligent Visual Support for Programming

In the VL/HCC (IEEE Symposium on Visual Languages and Human-Centric Computing) community, we can see a large amount of research addressing the issue that developers tend to interleave between activities like searching for relevant codes and collecting codes and other information that they believe would be necessary for editing or duplication (Ko, Myers, Coblenz, & Aung, 2006). These tools include navigational shortcuts to the code in IDE (Singer, Elves, & Storey, 2005), leveraging version history data to predict code changes (Zimmermann, Zeller, Weissgerber, & Diehl, 2005), better use of API (Stylos & Myers, 2006), and integration of web search or recommending source code examples in development environment (Brandt, 2010; Holmes & Murphy, 2005; Hsiao, Li, & Lin, 2008; Stylos & Myers, 2006). 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 (Chi, Pirolli, & Lam, 2007), such as online collaborative programming (social coding in GitHub[1]), Q&A websites, crowdsourcing suggestions, etc. (Bacchelli, Ponzanelli, & Lanza, 2012; Dabbish, Stuart, Tsay, & Herbsleb, 2012; Goldman, Little, & Miller, 2011; Hsiao et al., 2008; Mujumdar et al., 2011; Nasehi, Sillito, Maurer, & Burns, 2012; Treude, Barzilay, & Storey, 2011; Vasilescu, Serebrenik, Devanbu, & Filkov, 2014). However, almost all of these tools are targeted at problem-solving augmentation, reducing coding cognitive overhead when coding, and utility features enhancement (i.e. collaboration). Tools to support learning activities are less evident.

3. TOPIC FACET MODEL

Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003) is an unsupervised algorithm that uses bag of words approach to perform statistical topic modeling, which is a well-established method for uncovering hidden structures in large text corpora. There are several variations of LDA-based topic models to successfully encapsulate large text semantics into topic words, such as online reviews, political opinions, microblog streams, email summaries etc. (Jo & Oh, 2011; Lan, Buntine, & Huidong, 2010; Liu et al., 2012; Wang, Agichtein, & Benzi, 2012). In this work, we present a novel Topical Facets Modeling (TFM) method to capture online forum posts semantics.

The TFM algorithm automatically detect topics from conversational and relatively short amount of texts in each forum post. It is an extension of LDA (Blei et al., 2003) and SLDA (Lan et al., 2010). A topic is a multinomial distribution of words that represents a concept from each forum post. A facet is a multinomial distribution of words that represents a more specific topic in the forum, for instance, extends (a java keyword) is one of the main facets in determining whether a program implemented inheritance concept in Java programming language or not. Thus, Topic Facet Model firstly adopts SLDA (Lan et al., 2010) in the topic model. Essentially, SLDA takes into account the position of each individual word of topic inference. It then forces all words in a sentence are generated from one topic. When a post is topic-specific, short-and-sweet, such as how to write a for loop?, SLDA is supposed to distinctively generate the corresponding topic word - loops. However, as we discussed earlier, an open discussion forums often mix with various complexities of posts. For instance, "Can an array of objects be iterated in enhanced for loop". Given the sentence

[1]https://github.com It is an online software repository site, which allows distributed revision control and source code management.

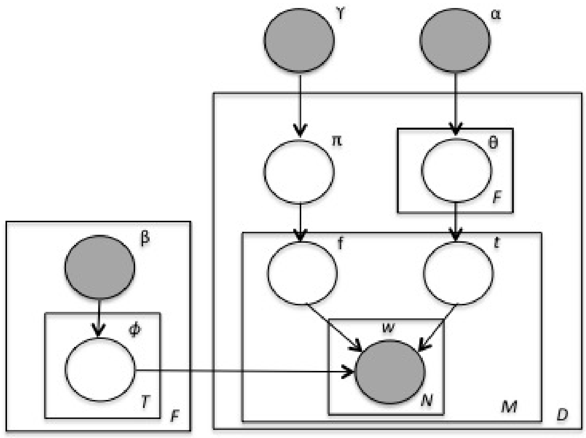

Figure 1: Topic Facet Model.

combines two main concepts, arrays and loops, SLDA will constrain only one topic word to be generated. In this case, the key of the question is about topic arrays (whether one can perform a function with array data structure), however, due to that there are more topic loops related words represented, the SLDA will misinterpret it. This is where the facets come into play, to take into account specificity of a topic in the model. Following the same example, we can specify 'array iteration' as a facet for topic loops (Hsiao,I-H, & Awasthi,P. 2015, to be appeared) . To explain Topic Facet Model algorithmically, Figure 1 shows the plate diagram. The words generative process is explained following.

1. For every pair of topic word t and facet f, draw a word distribution $\phi_{\mathrm{ft}} \sim$ Dirichlet (β_{f})
2. For each document d,

   a. Draw the document's topic word distribution $\pi_{\mathrm{d}} \sim$ Dirichlet (γ)

   b. For each topic word t, draw a facet distribution θ_{df} $\sim$ Dirichlet (α)

   c. For each sentence,
      - Choose a topic word j $\sim$ Multinomial (π_{d})
      - Given topic word j, choose a facet k $\sim$ Multinomial (θ_{dj})
      - Generate words w $\sim$ Multinomial (ϕ_{jk})

4. INSIGHT

In order to provide dynamic intelligent & personalized support for large-scale of online discussion forums, we build a web application and called it INSIGHT (since, it provides an insight on the concepts on which the answer has been built, to help user map his way to proper understanding of it), by using Django, Python and Javascript. The web application (Figure 2) re-structures a discussion forum site into 3 parts: Filters, Analytics Visualizations and Forum Posts. They are represented in the following three UI panels from left to right:

- **Control Panel (Left) -** contains a Search, three links - Inheritance, Loops, Stackoverflow.com. Inheritance and Loop links refreshes the section 3 with the respective posts data. It also refreshes the section 2 TFM bubbles. The search bar performs a normal search against the data on the keywords fed into it.

- **Analytics panel (Top Right) -** This section provides the result of our TFM model on the data provided in section 3. The results are showcased in form of bubble chart with some words mentioned in the center of each bubble. These words in the bubble chart are the most highlighted topics discovered by our algorithm. The size variation of individual bubbles defines the topic word relevance to the data i.e. Bigger the circle, bigger is relation of data to that topic.

  The bubbles are sectioned into two different color codes - one showing the topic related to the data and the other showing topics which are not related. Our TFM model clearly detects these differences, we call them facets and non facets. The bubble chart also changes according to the link selected on the left i.e. inheritance and loop. For inheritance the bubbles show following TFM facets - class, inheritance, extend, multiple, implement and following non-facets - if, call, type, composite, problem, which. For loops the bubbles show following TFM facets - for, do, loop, instance and following non facets - time, compile, value, optimism, variable. The TFM facets individually are clickable and work like a tag selection. On click, the data in section 3 gets sorted in descending order on the TFM value of the bubble clicked.

- **Forum posts (Bottom Right) -** contains all the forum posts data (question, its accepted answer (if available) and the next top voted answer) on the topic chosen in control panel, i.e., inheritance or loop. Each post contains some text and code, if available. In addition to the texts, each row also contains the TFM bubble, again, the size of the bubble denoting the facet relevance to the content of the post. The purpose of associating each post with its TFM facets value is to help users browse faster to find the related question to their problem. Every row of question is expandable. Once the user finds a related question to his problem, he can click on it to reveal its answer.

INSIGHT has been developed on Django and Python (interpreted languages) therefore, it can be scaled for larger data sets without compromising on the processing time. The architecture of the application has also been optimized to handle larger data sets. Moreover, all the visualization on the application is handled by javascript, therefore, providing the facility to incorporate more chart visualization like d3.js without worrying on the cost of efficiency, as these are

Table 1: Topic Facet Model notations

D: number of posts, M: number of sentences, N: number of words, T: number of topic-words, F: number of facets,; ω: word, t: topic-word, f: facet, ϕ: multinomial distribution over words, θ: multinomial distribution over topic-words, π: multinomial distribution over facets, : Dirichlet prior vector for θ, $\beta(_{\mathrm{w}})$, $\beta_{\mathrm{j(w)}}$: Dirichlet prior vector for ϕ(of facet j), $\gamma_{\mathrm{(j)}}$: Dirichlet prior vector for π

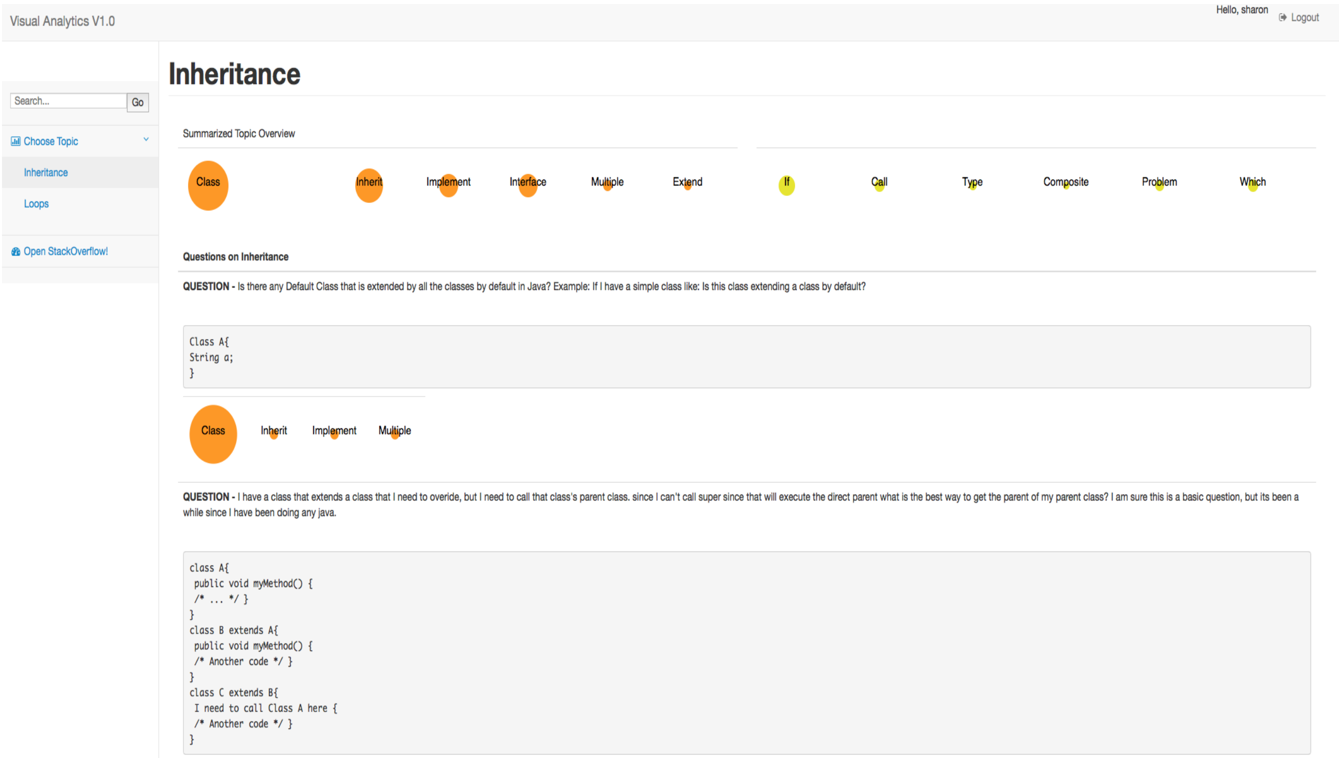

Figure 2: Interface of INSIGHT.

well optimized javascripts designed to handle data sets of any size.

4.1 The Analytics

4.1.1 *Implementation*

We implemented user tracking using Javascript on INSIGHTS. Javascript offers a quick and easy way to collect aggregate data on users and is built into INSIGHT. The system as a whole is a comprehensive logging system that tracks user's actions to a specific session. We also wish to provide a debrief of the session to user for improving his learning.

There are multiple third party tracking tools for example google analytics. But they all lack the ability to track an individual user's actions/decisions in chronological order. For example, you could see user clicked on question 1 and 5 to formulate his answer but with GA you cannot determine whether question 1 was clicked first or the question 5. Also, GA doesn't provide all the analytics together and it requires to be combined with other analytical tools to provide the full comprehensive logging system.

Because the order of actions is especially crucial in analytics, we built a new feature for INSIGHT to track all of user's actions in a log, which includes several pieces of information:

- What action they performed
  1. Mouse click on the page
  2. Scrolling up or down
  3. Which buttons were clicked
  4. What text were highlighted
  5. Which keywords were used in search bar
  6. Which questions were expanded for answers
  7. Which TFM bubble did the user click on to sort the data.
- When they performed the action
- On which page they performed the action

Furthermore, we added a tracking feature on User study page as well for all the decisions that the user makes during answering the questions. This feature tracks when a question was answered and which question was answered first.

All of the information from in-application actions and from the user study page are recorded continuously throughout the session. The data is stored for further aggregate analysis and research.

4.1.2 *Benefits and applications*

The tracking/logging feature built in the application allows us to drill down and filter by any of the levels, so we can easily see which actions were performed on which page and at what time. This also allow us to identify the common mistakes and patterns users follow during finding an answer to his coding problem. These mistakes or patterns are then to be addressed with further analysis and research and then built into the application to improve user learning.

The data can also be filtered over date, so further analysis can also be done to study the change of user's understanding over time, which may also be correlated with improvements in learner's knowledge.

5. USER STUDY

5.1 The Design

Table 2: Study Design

Topic 1 - Loops	Topic 2 - Inheritance
Experiment	Control 1
Experiment	Control 0
Control 1	Experiment
Control 0	Experiment

The user study has been designed to test the functional application of INSIGHT and its efficiency against other public online forums. For this particular case study, we use 'Stackoverflow.com' to do comparative study. Table 2. displays our study design with four sets of control environment to test the application thoroughly.

Table defines three control groups - experiment, control 1 and control 0.

- **Experiment** - user will answer the question of the respective section using INSIGHT.
- **Control 1** - user will answer the question of the respective section using help from Stackoverflow.com.
- **Control 0** - user is not allowed to refer Stackoverflow.com for solving the problem.

Though the experiment group and control 0 group use the same approach to answer the problem in hand i.e. referring to visual analytics for help, they have been defined as different set as the experiment group will access programming help through visual analytics interface only and the control group may get access to programming help through Stackoverflow.com depending upon which control group it refers to.

5.2 Study Procedure

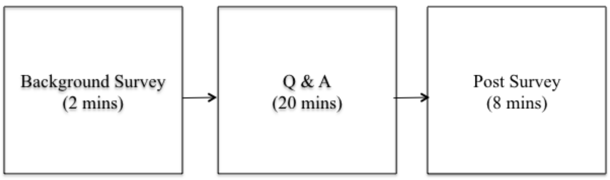

Figure 3: User study process flow.

Figure 3 shows the flow of the study. Every user is asked to go through the following three stages of the study -

- Background Survey
- Q&A
- Post Survey

Background survey is all about knowing the user's knowledge level in the area of coding and also involves asking him how well he is versed with online help i.e. does he uses google and stackoverflow and if he does then how well he is involved in the process.

The Q& A involves asking user two questions - one on topic loops and the other on topic inheritance. The user is required to answer both of the questions by referring to INSIGHT or stackoverflow.com (depending upon which control group the user belongs to). This task is time bound with 10 minutes allotted to each individual question adding it up to 20 minutes in total.

Post completion of the questions, the user is asked to provide a post survey to get their feedback to help improve the application. Post survey is the normal feedback system with users being asked to rate our application on the scale of 1 to 5 (where, 1-very bad and 5-excellent) on satisfaction, ease of use, ease of learning and usefulness. Users are also provided a space to give any comments on how can the application be improved. All of this data with user actions are stored in the database and later will be used to run more experiments using hidden topic markov model to find out how constructive the user responses are(Jeong, Gupta, Roscoe, Wagster, Biswas, Schwartz)

6. EVALUATION

6.1 Data Collection

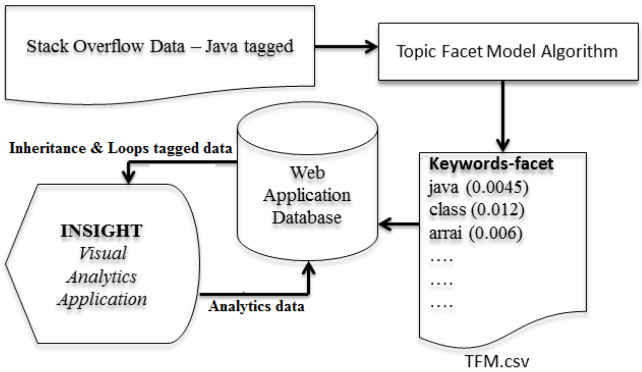

Figure 4: Data Collection.

We sampled one year (year 2013) of forum posts in topic Java from stackoverflow site through StackExchange API. The data pool was selected from the top 10 frequent tagged questions due to most of the posts in this section contained at least one accepted answer. For our case, we only show top 2 frequent tagged out of those 10 i.e. 'Inheritance' and 'Loops'. It will allow us to build a baseline to test INSIGHT on smaller set of data and also it's effectiveness. Later, the application will be scaled up to include all frequent tagged topics and questions.

6.2 User-Study Evaluation

INSIGHT version 1 prototype was recently developed and there are many use cases and further user studies underway. Till now, we have conducted 4 user study testing all four combinations of control environment as shown in Table 2. As the number of users were limited, we provide a qualitative evaluation of our application.

On the base of the background information provided by the users, the users can be clearly divided into two sets - 1. Users with some programming experience and 2. Users with no programming experience. Each set contained 2 users each and these sets were formed completely on the basis of how well they knew coding and how well they are familiar with the online coding forums.

1. How to break out of nested loops?

Figure 5: Loops Question.

The users were presented with same set of two questions - a easy problem on topic loops and a slightly difficult problem on topic inheritance. Figure 5 and 6 shows the snippet of the questions.

2. Review following code and answer the question below -

```
Class A{
        A(int a, int b){
                System.out.println("Hello");
        }
}

Class B extends A{
        B(){
                System.out.println("Class B");
        }
}
```

Output: Compile Time error - No constructor matching A(a,b) found in class B.

Why?

Figure 6: Inheritance Question.

Based on the control group users were required to access the respective resources and answer the questions. Out of the four, two users were able to answer both the question, whereas other two were only able to answer question on topic loops. Moreover, the two users who were able to answer both the question were the users who had some background knowledge of coding and were involved in some online discussion forums i.e. set-1. Because both the set involved one individual case, where the user was allowed to access 'Stackoverflow.com' i.e. Control 1, it comes as no surprise that users with some background knowledge were easily able to browse through the resources (Stackoverflow.com & INSIGHT) and find the solution and the other users weren't.

User's who failed to answer the problem on inheritance belonged to set-2 i.e users with no background knowledge. These users were not able to find the solution either on Stackoverflow.com or on INSIGHT, which provides an intuition whether users require some background knowledge to find the solution or not. It also points to INSIGHT being not so helpful for the users to find solution for the inheritance question. This intuition and deduction has been followed on very small dataset therefore it provides no concrete evidence for whatsoever. To testify for the intuition we require more rigorous testing and user studies.

During the study, users used search and TFM tagging facility extensively to find answer to the questions. Users found TFM bubble chart helpful as it helped them browse through the questions faster. The variation in the sizes of it assisted them to relate to the relevance of the question more easily. Search bar of the application worked in supplement with the TFM bubble chart to help users find related questions, hence the solution. The most frequent words searched for the question loops were - 'break', 'loops' and for question inheritance - 'extends', 'compilation error'.

Though the users liked the ease of use of the application, they felt the need of visually improving the application on the same lines. Collectively, INSIGHT was positively received by the users.

7. FUTURE DIRECTIONS & DISCUSSIONS

With so many variations and wideness in teaching style and technology, finding out ways to make learning effective and interesting becomes quite a task. Here are some of the ways we can lead INSIGHT in directions to make it more personalized:

INSIGHT logs and stores all the user's action on it with individual timestamp of when they were performed. We can filter this data by date, so further analysis can be done over the change of providing an answer by respective users, which may also be correlated with improvement in user's knowledge. Also, providing a dashboard for individual users to track or debrief on there performance by reviewing their answer and action logs can help users to gain deeper insight into their conceptual learning level and also help them review what data they used to formulate the answer. There logs can also be then used for identifying the area of weakness and then can be used by to provide more personalized help.

An ability to provide and instant feedback based on the user's action is very conducive to the improvement in user's knowledge or learning. This will also help users to form a empathetic connection as providing instant feedback stimulates a gesture of more personalization.

8. LIMITATIONS

In this paper we describe a functional prototype of visual analytics tool - INSIGHTS for discourse centric content. Our preliminary results demonstrated that INSIGHTS could be a promising approach to help users really learn and understand the concepts instead just writing the answers to problems but there are several limitations in current implementation. 1) The current version has been only tested for two topics - Loops and Inheritance out of 10 topics that were explicitly chosen to represent Easy & Difficult topics for CS1 course. 2) The user study was only conducted with limited subjects and requires more rigorous testing of our application and also to aggregate quantifiable results to test our hypotheses. 3) We currently only experimented the bubble charts visual representation on the extracted content semantics. We completely ignored the semantic relations, such as the concept causal relations, sequential or network visualizations.

9. REFERENCES

[1] V. Aleven, B. Mclaren, I. Roll, and K. Koedinger. Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education*, 16(2):101–128, 2006.

[2] J. R. Anderson and E. Skwarecki. The automated tutoring of introductory computer programming. *Communications of the ACM*, 29(9):842–849, 1986.

[3] K. E. Arnold. Signals: Applying academic analytics. *Educause Quarterly*, 33(1):n1, 2010.

[4] R. K. Atkinson and A. Renkl. Interactive example-based learning environments: Using interactive elements to encourage effective processing of worked examples. *Educational Psychology Review*, 19(3):375–386, 2007.

[5] A. Bacchelli, L. Ponzanelli, and M. Lanza. Harnessing stack overflow for the ide. In *Recommendation Systems for Software Engineering (RSSE), 2012 Third International Workshop on*, pages 26–30. IEEE, 2012.

[6] T. Barnes and J. Stamper. Toward automatic hint generation for logic proof tutoring using historical student data. In *Intelligent Tutoring Systems*, pages 373–382. Springer, 2008.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[8] K. E. Boyer, R. Phillips, A. Ingram, E. Y. Ha, M. Wallis, M. Vouk, and J. Lester. Investigating the relationship between dialogue structure and tutoring effectiveness: a hidden markov modeling approach. *International Journal of Artificial Intelligence in Education*, 21(1):65–81, 2011.

[9] J. Brandt, M. Dontcheva, M. Weskamp, and S. R. Klemmer. Example-centric programming: integrating web search into the development environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 513–522. ACM, 2010.

[10] E. H. Chi, P. Pirolli, and S. K. Lam. Aspects of augmented social cognition: Social information foraging and social search. In *Online Communities and Social Computing*, pages 60–69. Springer, 2007.

[11] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. Social coding in github: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1277–1286. ACM, 2012.

[12] K. Dave, 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*, pages 232–241. ACM, 2004.

[13] L. Du, W. L. Buntine, and H. Jin. Sequential latent dirichlet allocation: Discover underlying topic structures within a document. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 148–157. IEEE, 2010.

[14] M. Goldman, G. Little, and R. C. Miller. Collabode: collaborative coding in the browser. In *Proceedings of the 4th international workshop on Cooperative and human aspects of software engineering*, pages 65–68. ACM, 2011.

[15] B. Gretarsson, J. O'donovan, S. Bostandjiev, T. Höllerer, A. Asuncion, D. Newman, and P. Smyth. Topicnets: Visual analysis of large text corpora with topic modeling. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):23, 2012.

[16] B. V. Hanrahan, 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*, pages 91–94. ACM, 2012.

[17] R. Holmes and G. C. Murphy. Using structural context to recommend source code examples. In *Proceedings of the 27th international conference on Software engineering*, pages 117–125. ACM, 2005.

[18] I.-H. Hsiao, Q. Li, and Y.-L. Lin. Educational social linking in example authoring. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 229–230. ACM, 2008.

[19] I.-H. Hsiao, S. Sosnovsky, and P. Brusilovsky. Guiding students to the right questions: adaptive navigation support in an e-learning system for java programming. *Journal of Computer Assisted Learning*, 26(4):270–283, 2010.

[20] Y. Jo and A. Oh. Aspect and sentiment unification model. 2010.

[21] A. J. Ko, B. A. Myers, M. J. Coblenz, and H. H. Aung. An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks. *Software Engineering, IEEE Transactions on*, 32(12):971–987, 2006.

[22] Y.-J. Lee, E.-K. Kim, H.-G. Cho, and G. Woo. Detecting and visualizing online dispute dynamics in replying comments. *Software: Practice and Experience*, 43(12):1395–1413, 2013.

[23] S. Liu, M. X. Zhou, S. Pan, Y. Song, W. Qian, W. Cai, and X. Lian. Tiara: Interactive, topic-based visual text summarization and analysis. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):25, 2012.

[24] S. Y. Lye and J. H. L. Koh. Review on teaching and learning of computational thinking through programming: What is next for k-12? *Computers in Human Behavior*, 41:51–61, 2014.

[25] D. Mujumdar, M. Kallenbach, B. Liu, and B. Hartmann. Crowdsourcing suggestions to programming problems for dynamic web development languages. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 1525–1530. ACM, 2011.

[26] S. M. Nasehi, J. Sillito, F. Maurer, and C. Burns. What makes a good code example?: A study of programming q&a in stackoverflow. In *Software Maintenance (ICSM), 2012 28th IEEE International Conference on*, pages 25–34. IEEE, 2012.

[27] C. Piech, M. Sahami, D. Koller, S. Cooper, and P. Blikstein. Modeling how students learn to program. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 153–160.

ACM, 2012.

[28] A. Robins, J. Rountree, and N. Rountree. Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2):137–172, 2003.

[29] S. B. Shum et al. Cohere: Towards web 2.0 argumentation. *COMMA*, 8:97–108, 2008.

[30] G. Siemens and R. S. d Baker. Learning analytics and educational data mining: towards communication and collaboration. In *Proceedings of the 2nd international conference on learning analytics and knowledge*, pages 252–254. ACM, 2012.

[31] M.-A. Storey. Theories, methods and tools in program comprehension: Past, present and future. In *Program Comprehension, 2005. IWPC 2005. Proceedings. 13th International Workshop on*, pages 181–191. IEEE, 2005.

[32] J. Stylos 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*, pages 195–202. IEEE, 2006.

[33] C. Treude, O. Barzilay, and M.-A. Storey. How do programmers ask and answer questions on the web?: Nier track. In *Software Engineering (ICSE), 2011 33rd International Conference on*, pages 804–807. IEEE, 2011.

[34] J. Tsui, S. EdTech, and J. Benfield. Brief. 2014.

[35] C. van de Sande and G. Leinhard. Online tutoring in the calculus: Beyond the limit of the limit. *education*, 1(2):117–160, 2007.

[36] T. VanDeGrift. Coupling pair programming and writing: learning about students' perceptions and processes. *ACM SIGCSE Bulletin*, 36(1):2–6, 2004.

[37] B. Vasilescu, A. Serebrenik, P. Devanbu, and V. Filkov. 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*, pages 342–354. ACM, 2014.

[38] J. Vassileva, C. Gutwin, et al. Exploring blog archives with interactive visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pages 39–46. ACM, 2008.

[39] K. Verbert, E. Duval, J. Klerkx, S. Govaerts, and J. L. Santos. Learning analytics dashboard applications. *American Behavioral Scientist*, page 0002764213479363, 2013.

[40] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. Tiara: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162. ACM, 2010.

[41] A. F. Wise, Y. Zhao, and S. N. Hausknecht. Learning analytics for online discussions: a pedagogical model for intervention with embedded and extracted analytics. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 48–56. ACM, 2013.

[42] T. Zimmermann, A. Zeller, P. Weissgerber, and S. Diehl. Mining version histories to guide software changes. *Software Engineering, IEEE Transactions on*, 31(6):429–445, 2005.