Computers in Human Behavior 72 (2017) 771–782

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Full length article

Enriching programming content semantics: An evaluation of visual analytics approach

I-Han Hsiao [a, *], Yi-Ling Lin [b]

[a] School of Computing, Informatics & Decision Systems Engineering, Arizona State University, 699 S. Mill Ave., Tempe, AZ 85281, USA
[b] Department of Information Management, National Sun Yat-Sen University, No. 70, Lienhai Rd., Kaohsiung 80424, Taiwan

ARTICLE INFO

Article history:
Received 1 May 2016
Received in revised form
9 October 2016
Accepted 12 October 2016
Available online 25 October 2016

Keywords:
Semantic indexing
Automatic indexing
Visual analytics
Interactive visualization
Learning analytics
Programming learning

ABSTRACT

In this work, we present an intelligent classroom orchestration technology to capture semantic learning analytics from paper-based programming exams. We design and study an innovative visual analytics system, EduAnalysis, to support programming content semantics extraction and analysis. EduAnalysis indexes each programming exam question to a set of concepts based on the ontology. It utilizes automatic indexing algorithm and interactive visualization interfaces to establish the concepts and questions associations. We collect the indexing ground truths of the targeted set from teachers and experts from the crowd. We found that the system significantly extracted more and diverse concepts from exams and achieved high coherence within exam. We also discovered that indexing effectiveness was especially prevalent for complex content. Overall, the semantic enriching approach for programming problems reveals systematic learning analytics from the paper exams.

1. Introduction

Paper-based exams are one of the main assessment methods in today's majority of classrooms. Such delivery method is especially beneficial for the sake of easiness in exam-proctoring and preventing academic dishonesty. However, they are in fact very time-consuming to grade, hard to maintain consistency among graders, and normally contain only very limited feedback to a student. Furthermore, it is impractical for an instructor to track detailed performance of a student (e.g., how s/he received partial credits in different exam questions), instead, teachers discuss on the returned exam in the class (hopefully thorough and detailed enough to cover all the students' misconceptions). Although teachers may still point out the common mistakes and try to pinpoint the key concepts related to the such mistakes during instruction, many desired detailed learning analytics are unavailable, such as how did s/he receive partial credits, was it a single concept or multiple concepts mistake, a careless mistake or a long-term misconception etc. As a result, students often focus solely on the scores they earned on the returned exams, but miss several learning opportunities (Ambrose, Bridges, DiPietro, Lovett, & Norman, 2010) such as *identification of strength and weakness, characterization of the nature of their errors or any recurring patterns if any, assessment of appropriateness of their study strategies and preparation*, etc. Hence, making it impossible to apply learning analytics for delivering personalized feedback to the student. Therefore, unlike most of the orchestration technologies, which mainly address digital form of educational data (Dillenbourg, 2013), in this work, we propose an educational technology solution that permits traditional paper delivery method to be able to utilize advanced learning analytics by analyzing the textual content and supplying semantic information.

In order to provide additional learning analytics for traditional paper-based exams in facilitating today's majority classes, we focus on a targeted domain, programming language learning, and a targeted paper delivery content, paper-based programming exams. We create an innovative visual analytics system, EduAnalysis, to analyze the content and to index it to a set of concepts based on the ontology. EduAnalysis implements an automatic indexing algorithm and interactive visualization interfaces to establish the concepts and exam questions semantic associations. Our core research question is whether the proposed approach can effectively capture advanced programming learning analytics to enhance paper-based programming assessments. Specifically, we hypothesize that the indexing method can provide richer information to the content and the indexing approach can facilitate content analysis in traditional

* Corresponding author.
E-mail addresses: Sharon.Hsiao@asu.edu (I.-H. Hsiao), yllin@mis.nsysu.edu.tw (Y.-L. Lin).

http://dx.doi.org/10.1016/j.chb.2016.10.012

paper-based programming assessments. To verify these hypotheses, we collect the indexing ground truths from teachers and experts from the crowd and compare the results with the proposed algorithmic method.

The main contributions of this work are outlined as following:

- Provide immediate technology support for today's majority programming classes, particularly (large) blended instruction classrooms that are instrumenting paper-based formal assessments;
- Introduce novel intelligent semantic parser to automatically associate concepts and programming problems;
- Present visual authoring, delivery and presentation interfaces via semantic analytics visualizations in the targeted context;
- Conduct controlled crowdsourcing experiment to harness educational ground truth;
- Empirically evaluate the proposed intelligent semantic indexing method to address a real world problem.

The rest of the paper is structured with literature review on topics *classroom orchestration & learning analytics, semantic enrichment to enhance learning and visual learning analytics & student modeling*. In section 3, we present the visual analytics system, EduAnalysis. In section 4, we lay out our study methodology with our underlying assumptions and evaluation measures. Finally we present the evaluation results and discussed study implications, limitations and future work.

2. Literature review

2.1. Orchestration & learning analytics

In the field of Computer Supported Collaborative Learning (CSCL), researchers describe course-delivery as a field in transition for classroom orchestration, which defines how a teacher manages multilayered activities in real time and in a multi-constraints context (Dillenbourg, 2013). Orchestration emphasizes attention to the challenges of classroom use and adoption of research-based technologies (Roschelle, Dimitriadis, & Hoppe, 2013). It discusses how and what research-based technologies have been adopted and should be done in classrooms (Dillenbourg, 2013). We have begun to see more tabletops, smart classrooms or interactive tools such as Classroom Response Systems (AKA: Clickers) etc. provide dynamic feedback and integrative students knowledge updates (Martinez-Maldonado, 2014; Martinez-Maldonado, Dimitriadis, Martinez-Monés, Kay, & Yacef, 2013; Roschelle, Penuel, & Abrahamson, 2004; Slotta, Tissenbaum, & Lui, 2013). One of the biggest criticisms of introducing orchestration technology in class is that it might potentially add more complexity and time demands of technology and introduces new and unnecessary complications (Sharples, 2013). Thus, it motivates us to research a less intrusive technological solution that taps into blended classes allowing to manage physical and digital content and to jointly discuss learning analytics.

Vatrapu, Teplovs, Fujita, and Bull (2011) describe a preliminary framework, Triadic Model of Teaching Analytics (TMTA), discussing the importance involving three stakeholders in learning analytics: teaching expert, visual analytics expert and design-based research expert. The focus of learning analytics has been on the integration of computational and methodological support for teachers to properly design, deploy and assess learning activities. In addition, the focus is also to immerse students in rich, personalized and varied learning activities in information ecologies and data-rich classrooms (Vatrapu et al., 2011). One of the pioneer systems that align with TMTA framework is eLab (exploratory Learning Analytics Toolkit). It was designed to enable teachers to explore and correlate content usage, to help teachers reflect on their teaching according to their own interests (Dyckhoff, Zielke, Bültmann, Chatti, & Schroeder, 2012). ASSISTments (Heffernan & Heffernan, 2014) an integrative tutoring system includes assistance and assessment components for students and teachers. The system is built on a mantra - put the teacher in charge, not the computer, which creates flexibility to allow teachers to use the tool in organizing the classroom routines. However, such intelligent tutors or newly invested orchestration technologies are typically highly customized to the content or require a large collection of content for teachers to start using the tools. In this work, we propose and evaluate an automatic method to enrich content semantics in bridging physical and digital via visual learning analytics.

2.2. Semantic enrichment to enhance learning support

Semantic approaches have been widely discussed in current computer-based education. There is a line of ontology related studies being pursued by a number of researchers in different aspects of learning, such as learning content authoring and management, contextual annotation and support, personalized search and content composition, learning resource and metadata repositories (Tiropanis, Davis, Millard, & Weal, 2009), etc.

AIMS (Aroyo & Dicheva, 2001) and TM4L (Topic Maps for Learning) (Dicheva, Dichev, Sun, & Nao, 2004) are two good examples for contextual annotation and support. They both enable learners to identify related information resources for different tasks such as course assignments. They provide the complementary support for learning tasks through subject domain conceptualization methods. The project of iHelp Presentation (Bateman, Brooks, Mccalla, & Brusilovsky, 2007) helps learners to highlight important parts of the recorded lectures' slides and support them tagging, annotation, and collaboration features around the recordings. Research conducted by the LORNET network (Paquette, 2007) offers a semantic framework to manage the learners' competency portfolios and models in e-learning and knowledge management environments. The work presented in (Jovanovic et al., 2007) demonstrates the semantic technologies enable a generic implementation of feedback for content authors and teachers to aware about the quality of the learning process based on students' activities in online learning environments. ArnetMiner team (2008) developed the system at extracting and mining academic social networks to expertise search and people association search. Alomari, Hussain, Turki, and Masud (2015) developed a semantic model for collaborative learning by graphically representing course content with semantic meaning.

Assessment in learning can be characterized as an index of learning guidance or a summary of learners' performance (Basu, Jacobs, & Vanderwende, 2013). In the context of automatic evaluation, there is a stream of research focuses on the correctness of syntactical references by using pattern-matching techniques to verify solutions (i.e. WEB-CAT auto-grading (Edwards & Perez-Quinones, 2008). There are other streams of work that emphasizes on semantic relations, such as TagAssessment (Kardan, Sani, & Modaberi, 2016), which has been proposed for assessing leaners by computing the semantic relationship between educational contents and learner's tags on multiple choice questions (MCQ). Mohler and Mihalcea (2009) applied various measures of lexical similarity based on WordNeT and Latent Semantic Analysis(LSA) to automatic short answer grading. Basu et al. (2013) introduce a semi-automatic grading approach to allow teachers to grade easily with fewer actions, provide feedback to groups of similar answers, and discover modalities of students' misunderstanding. Including our current work, we design a visual analytics system that utilizes automatic indexing algorithm and interactive visualization interfaces to analyze paper-based programming content semantics. Such tool assists programming instructors to be able to continue instrumenting traditional paper-based formal assessments and be able to obtain semantic learning analytics.

2.3. Visual learning analytics & student modeling

Visual learning analytics, essentially, extends the scope of information visualization by using computer-supported techniques to visualize learning information in amplifying human cognition. It goes beyond the "footprints" representation of summarizing and visualizing interactions or behaviors between students and learning content. Examples like network visualizations in semantic discourse analysis (De Liddo, Shum, Quinto, Bachler, & Cannavacciuolo, 2011), dashboard visualizations to provide historical data in supporting awareness, teaching practices, explore and/or identify monitor status (Epp & Bull, 2015; Verbert, Duval, Klerkx, Govaerts, & Santos, 2013). There is a range of visual learning analytics cases reported in VISual Approaches to Learning Analytics (VISLA) workshop, in the Fifth International Conference of Learning Analytics and Knowledge (Duval et al., 2015). For instance, applying sentence compression technique in analyzing short answer questions in network visualizations; utilizing predictive modeling to visualize uncertainty of academic risks; innovative visualizations for visualizing semantics in discussion forums (Awasthi & Hsiao, 2015) etc. Studies showed that the majority of visual learning analytics discusses visual representations or the system's usefulness while the core should be focused on real impact to improve learning or teaching (Verbert et al., 2013).

From student modeling literature, we found several successful examples presented interactive visualizations in supporting students' learning. Such approach is called Open Student Modeling (OSM). It is a group of approaches 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 complex concept maps or Bayesian 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 & Kay, 2016; Bull, 2004; Mitrovic & Martin, 2007; Zapata-Rivera & Greer, 2000). Several other examples of OSM interfaces reported promising results too. For instance, interacting with open learner modeling engages learners in negotiating with the system during the modeling process (Dimitrova, Self, & Brna, 2001). Progressor system integrates open learning models with social visualization that can dramatically increase student motivation to work with non-mandatory educational content (Hsiao, Bakalov, Brusilovsky, & König-Ries, 2013) and encourage students to start working sooner. Chen, Chou, Deng, and Chan (2007) investigated active open learner models in order to motivate learners to improve their academic performance. Both individual and group open learner models were studied and demonstrated the increase of reflection and helpful interactions among teammates. CourseVis provides various graphical representations of student tracking data to teachers and learners and helps instructors to identify problems early on, and to prevent some of the common problems in distance learning (Mazza & Dimitrova, 2007).

3. Research platform: EduAnalysis - semantic indexing and visual analytics

We build EduAnalysis, a semantic visual analytics system specifically designed to extract semantics from physical learning environment and map onto a virtual setup. EduAnalysis consists of algorithmic components for automation and interactive visualization interfaces for authoring. For a given scenario, a student studies course related e-textbook and slides online to prepare for an in class exam, works on online quizzes or other online materials, and eventually takes a paper-based exam or a quiz in classrooms. EduAnalysis aims to track all these types of assessment results and present the performance in visualization by harnessing the learning content semantics. The following section describes EduAnalysis' system architecture, design rationales and interfaces on how it enables creation of enriched formal assessments for the class, assimilates student data and present the staffs and students with a detailed semantic analysis of student performance.

3.1. EduAnalysis architecture

EduAnalysis was deployed as a web application. Fig. 1 depicts the architecture of EduAnalysis. There are three main components, frontend analytics dashboard and web services to process physical data input (such as paper exam processing service, manual concept indexing service etc.), backend consists of an ontology parser, a concept mapper that maps input content to their corresponding concepts, and an analytics framework that exposes insights from data using APIs, and output via dashboard. Modules that are colored in grey are currently still under development. EduAnalysis is built using MongoDB as (NoSQL) database, Python Flask as backend web server and AngularJS for frontend. The application is deployed on an EC2 instance. This paper focuses on the backend intelligence design and evaluation.

3.2. EduAnalysis interfaces

3.2.1. Indexing interfaces

Teachers can upload an exam paper with a simple one click (interface omitted) and EduAnalysis will trigger *ExamParser* service to perform automatic concept indexing and immediately lead teachers to an overview (Fig. 2). It guides teachers to navigate the entire exam concept distribution. Teachers can opt for further editing on exam questions or provide concept emphasis configuration. Fig. 3 shows a view of the authoring interface. Left panel displays each question texts, which enables dynamic editing and indexing to provide teachers instant feedback of the indexing performances. Three other parameters can be adjusted here: correct answer, corresponding marks, and question complexity. All these parameters are reserved for future *auto-grading services* and *partial credit assignment based on semantics* (Hsiao, 2016). Middle panel shows an interactive concept authoring circle packing visualization. Teachers can select the bubble to zoom in and out to examine the fine-grained concepts coverage. Fig. 3 also illustrates a zoomed in view of fully indexed question. By zooming in and out, teachers can select/deselect concepts for the corresponding question with a click on the bubble. They can also adjust the slider bar to configure the concept weights (emphasis).

3.2.2. Dashboard interfaces

Fig. 4 shows a dashboard view of class exam performance with three-layer information, including (a) a box & whisker plot of the class performance summary, (2) a stacked bar chart of class' exam scoring distribution by question, allowing to visualize the aggregated performance per question, in terms of correct/incorrect/partially correct/no answers with question complexity, and (3) detail student scoring by quarter. The visualization dashboard reflects higher-level feedback on the exam results. Note: sensitive data (students' names) is blurred in Fig. 4.

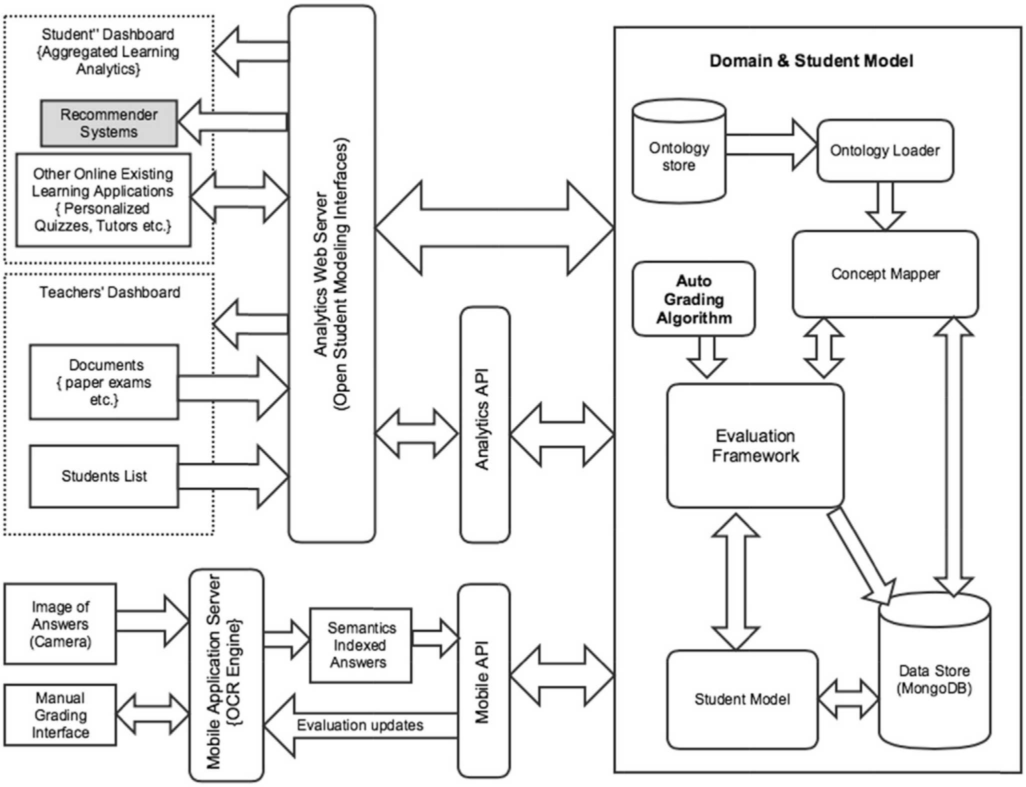

Student'' Dashboard {Aggregated Learning Analytics}
Recommender Systems
Other Online Existing Learning Applications { Personalized Quizzes, Tutors etc.}
Teachers' Dashboard
Documents { paper exams etc.}
Students List
Analytics Web Server (Open Student Modeling Interfaces)
Analytics API
Domain & Student Model
Ontology store
Ontology Loader
Concept Mapper
Auto Grading Algorithm
Evaluation Framework
Student Model
Data Store (MongoDB)
Image of Answers (Camera)
Manual Grading Interface
Mobile Application Server {OCR Engine}
Semantics Indexed Answers
Evaluation updates
Mobile API

Fig. 1. EduAnalysis architecture.

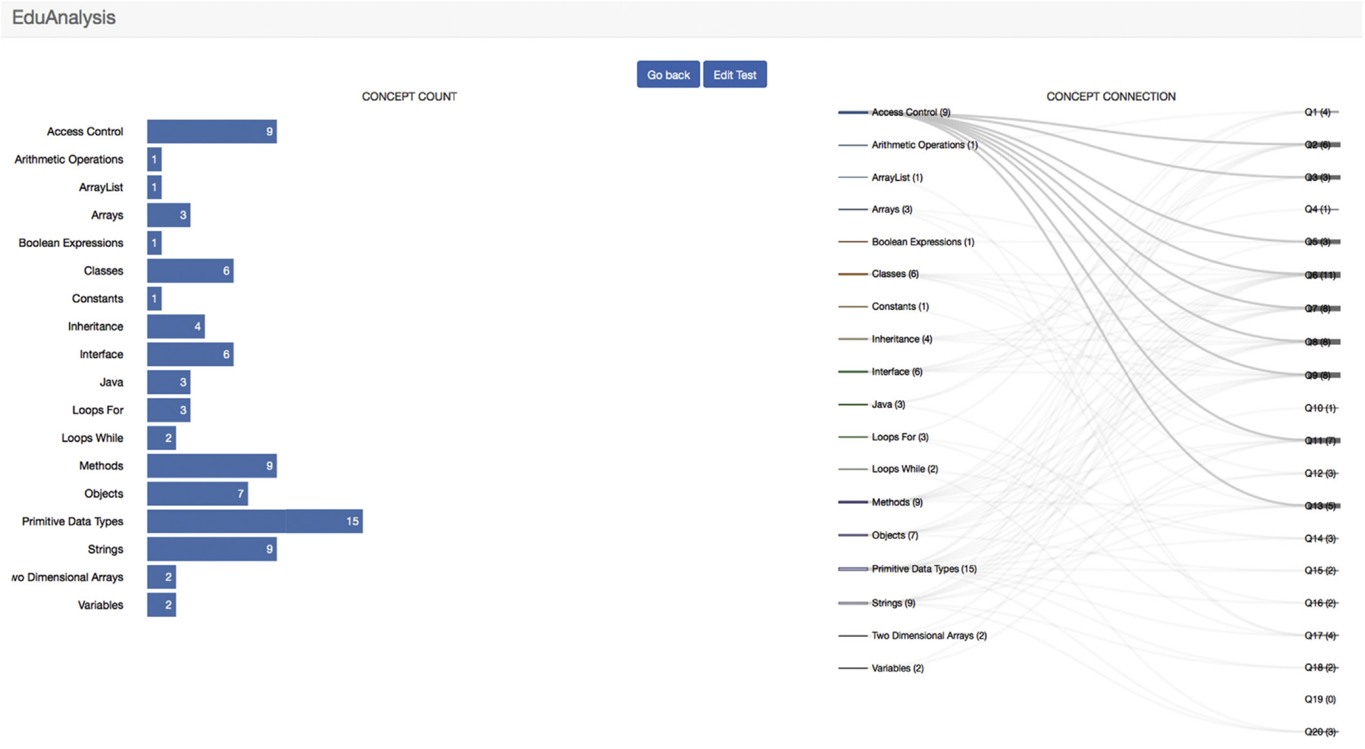
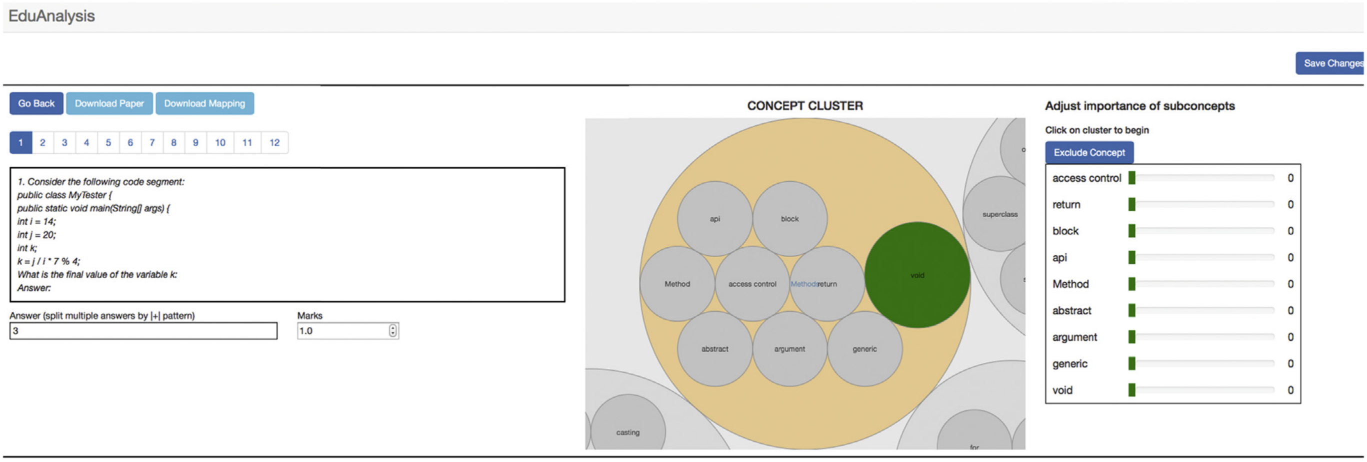


Fig. 2. Exam overview on topics and concepts distribution.

Finally, Fig. 5 shows the semantic analytics overview from teacher's perspective (Students will be seeing exactly the same semantic analytics with anonymized names from the same class. Teachers have a comprehensive overview of conceptual performances for the entire class, including three dimensions of concept performance: (1) a heat map sorted by topics: the color density of



1. Consider the following code segment:

```
public class MyTester {
public static void main(String[] args) {
int i = 14;
int j = 20;
int k;
k = j / i * 7 % 4;
```

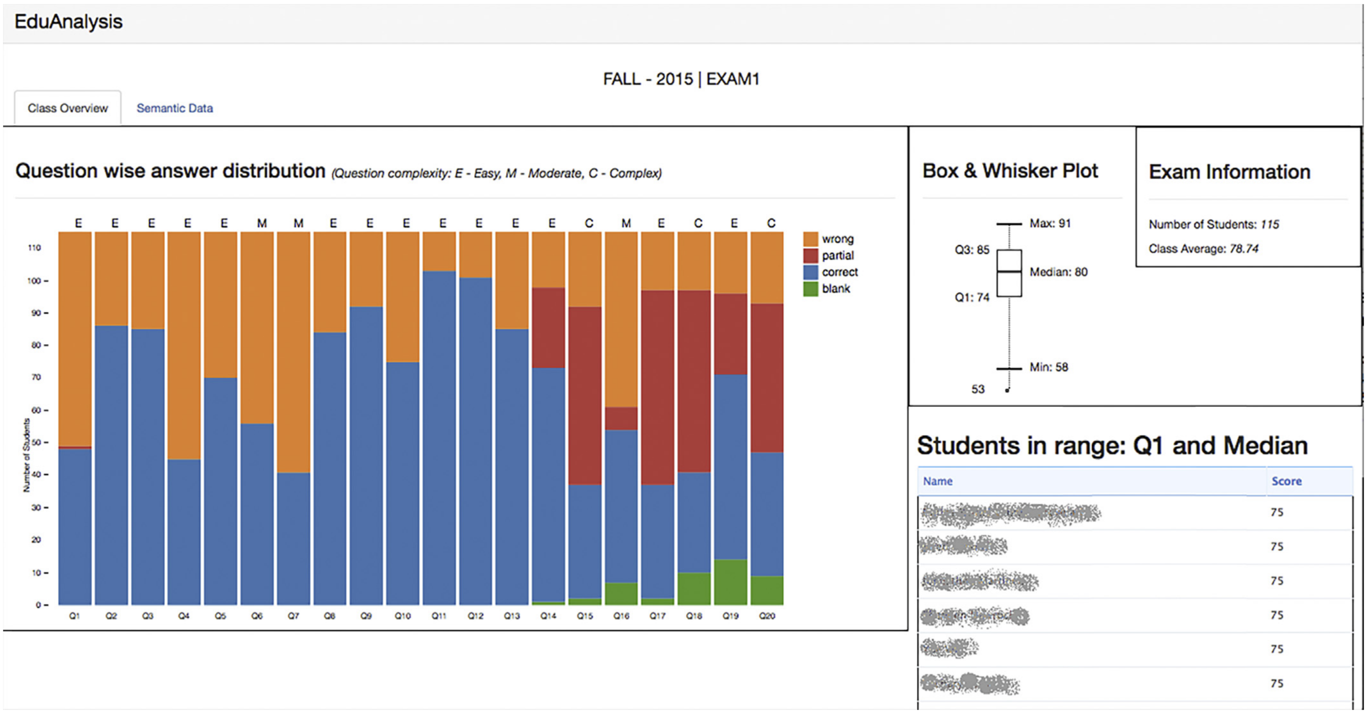
What is the final value of the variable k:
Answer:

Answer (split multiple answers by |+| pattern): 3

Marks: 1.0

access control	0
return	0
block	0
api	0
Method	0
abstract	0
argument	0
generic	0
void	0

Fig. 3. A navigational and authoring interface for question concepts. A full indexed question of question1 in exam1.



Exam Information

Number of Students: *115*

Class Average: *78.74*

Students in range: Q1 and Median

Name	Score
[redacted]	75
[redacted]	75
[redacted]	75
[redacted]	75
[redacted]	75
[redacted]	75

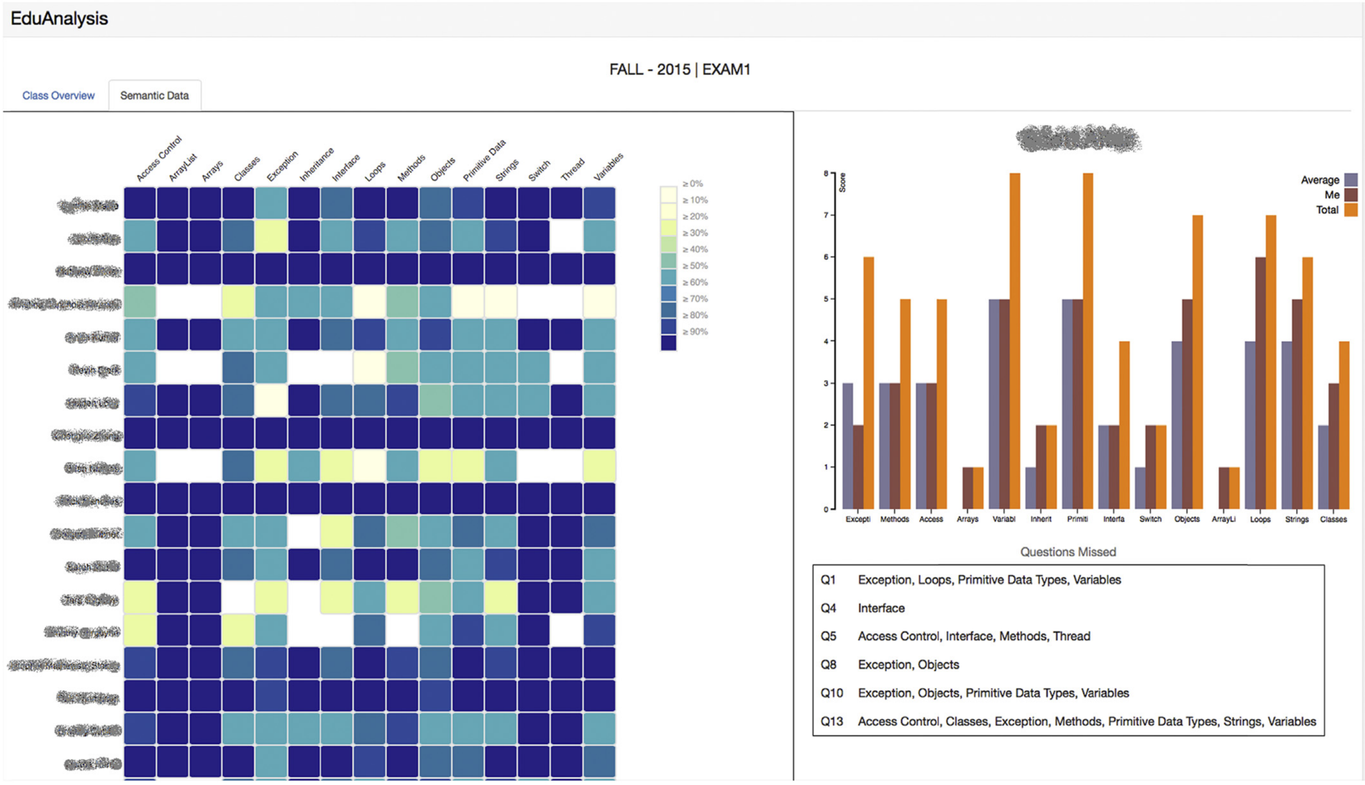
Fig. 4. Overall performance overview.

the grid indicates one's conceptual gain, the darker the more conceptual understanding of the topic; (2) a bar chart illustrating selected student's personal conceptual performances versus group average and overall goal, where the contrasting bars are intended to invoke user's goal setting and self-regulated learning and (3) lower-right corner of semantic overview shows the scrutiny of *questions-and-misconceptions* reference links, which displays the semantic feedback for every erroneous answer on the exam that typically paper-based exams are hard to do. Meanwhile, students have the consistent view as the teacher's with anonymous performances from his/her peer cohorts. The visualization design choices are made based on several successful open social student modeling interfaces design (Brusilovsky, Somyürek, Guerra, Hosseini, & Zadorozhny, 2015; Hsiao & Brusilovsky, 2012; Loboda, Guerra, Hosseini, & Brusilovsky, 2014).

4. Methodology

This project aimed to study the effectiveness of an intelligent support for semantic visual analytics and how teachers would perceive of using it in programming language courses. We hypothesized that intelligent automatic semantic indexer is an effective method to collect semantic information from course content.

We call the instance of automatic exam concept indexing service, *ExamParser*, it inherits from generic Topic Facet Model (Hsiao & Awasthi, 2015), which consists of natural language parser and domain specific language parser (in this project, we deployed Java Parser, which was originally developed by Hosseini and Brusilovsky (2013) and applied in (Hsiao, Sosnovsky, & Brusilovsky, 2010; Sosnovsky, Hsiao, & Brusilovsky, 2012), where the Java Ontology



Questions Missed

Q1	Exception, Loops, Primitive Data Types, Variables
Q4	Interface
Q5	Access Control, Interface, Methods, Thread
Q8	Exception, Objects
Q10	Exception, Objects, Primitive Data Types, Variables
Q13	Access Control, Classes, Exception, Methods, Primitive Data Types, Strings, Variables

Fig. 5. Teacher's semantic analytics view: personal vs. class concept performance heat map, bar chart of self performance versus average and overall goal. Note: sensitive data (students' names) is blurred.

can be retrieved here[1]). It recognizes exam question patterns in a document and extracts content by indexing each question to corresponding concepts (a high level concept topic and sets of facets). A typical exam question pattern includes:

question_text{phrased as natural language, may or may not contain domain specifics}

codes{composed as fully or partially of an entire executable program}

answer_type{ranges from multiple choices, fill-in-the-blanks, short answers, code writing etc.}

For instance, here's a sample exam question:

“What is the final value of sum displayed to the console?

```
for(int i = 0; i < 5; i++)
  {
     int sum = 0;
     sum = sum + i;
     System.out.println(sum);
  }”
```

This question contains mainly natural language phrased question descriptions, a piece of partial executable java codes, and multiple choices answer type for this question. *ExamParser* will translate this question as a set of concepts {*ForStatement, VariableInitialization, ConditionalStatement, LessOperator, IncrementOperator, MethodInvocation, AssignmentOperator, Arithmetics*}. However, do these concepts all weigh equally in this exam question? If we purely count the concept appearances, it consists of three *AssignmentOperators* and one *ForStatement*. Does it mean that *ForStatement* is less important than *AssignmentOperator* in this question? The answer is it depends! Therefore, we design a dynamic concept indexing authoring interface in the parser (Fig. 3). It labels each parsed concept with the equivalent, default quantity weights, but the weights are adjustable according to teachers' emphasis. In the case of using such question in a CS1 midterm exam, the focus should be on *ForStatement*, *ConditionalStatement* concepts; using the same question in a CS1 final exam, every concept should weigh equally proportionally. In addition, providing dynamic concept weight authoring interfaces not only allows teachers to include or exclude additional or redundant concepts to exam questions, but also enables dynamic exam content editing and corresponding concept indexing (Fig. 3). Embedding such dynamic authoring mechanism along with intelligent parsing can help raise teachers' flexibility to configure and coordinate entire exam topical emphasis, at the same time, complement to algorithmic flaws, in case of any missing concepts.

The concept indexing method enables a scalable framework in two essential educational technology aspects: (1) systematically assign partial credits, which they are traditionally provided by teachers' experiences or generic grading rubrics (such as credits to right path toward key concepts but erroneous implementation). By associating each programming problem to weighted concept sets facilitates an organized fashion to quantitatively distribute partial credits in semantic level (Hsiao, 2016); and (2) harness different levels of learning analytics on both individual and group levels, including strong and weak concept clusters, misconceptions co-occurrences, conceptual progress over time etc. In this paper, we focus on aggregating various levels of semantics analytics.

4.1. Data collection

We collected 4 programming introductory courses exams, with a total 76 exam questions in the subject of Object-Oriented

[1] Source of Java Ontology: http://www.pitt.edu/~paws//ont/java.owl.

Programming. Each exam was populated in **EduAnalysis**; each question was automatically associated with a set of concepts through *Topic Facet Model* algorithm (Hsiao & Awasthi, 2015).

Subject indexing is commonly used for the assignment of multiple subject terms to represent an item, such as a document, an image, or a problem. To verify the quality of indexing, indexing consistency has often been applied as a measure of the quality of a gold standard or as a measure of indexing quality without reference to a gold standard(Rolling, 1981). Medelyan and Witten (2006) proposed defining the "gold standard" in indexing as a level of inter-indexer consistency to evaluate a thesaurus-based indexing algorithm. In order to verify the embedded indexing algorithm effectiveness, we had collected two baselines of concept indexing for the targeted corpus (1) teacher judges, and (2) experts from the crowd.

4.1.1. Indexing by teachers

For teachers' indexing, we refer it to baseline I. There are two teachers, who both have more than 5 years teaching experiences in the subject domain. They manually examined every single exam question from the corpus by selecting concepts from a list of JAVA ontology and log the associations via spreadsheets. We later compile the indexed concepts from both spreadsheets and compute the inter-rater reliability, Cohen's Kappa = 0.386.

4.1.2. Indexing by experts from the crowd

Crowdsourcing has emerged as a popular approach to harvest collective wisdom from thousands of volunteers in different applications (Howe, 2008; Surowiecki, 2004). Classic crowdsourcing tasks are for various practices that involve the crowd via online platforms, such as correction, labeling, ranking, data cleaning, data filtering, data collection, and entity linking (Amirkhani & Rahmati, 2014; Demartini, Difallah, & Cudré-Mauroux, 2012; Franklin, Kossmann, Kraska, Ramesh, & Xin, 2011; Marcus, Wu, Karger, Madden, & Miller., 2011; Park et al., 2012). The approach can reduce costs dramatically by outsourcing certain process of the practice to the crowd rather than having the professionals perform all labor-intensive tasks.

Therefore, we designed a crowdsourcing indexing task to identify concepts of the given java programming questions for our study. The indexing task was designed in a survey style format on Qualtrics[2] platform and hosted on Amazon Mechanical Turks (MTurk).[3] Amazon's Mechanical Turk is an online labor market where requesters post jobs and workers choose which jobs to do for pay. Amazon's Mechanical Turk service has become an increasingly popular way to conduct online experiments (anyone with access to internet can use this service). Subjects on MTurk are from all over the world and are tagged with different levels of qualifications. Prior research has shown that respondents in MTurk might be slightly different from the respondents in a traditional subject pool, but the data obtained from MTurk were at least as reliable as those obtained by traditional methods (Buhrmester, Kwang, & Gosling, 2011; Paolacci, Chandler, & Ipeirotis, 2010). In this study, we designed three mechanisms prior work (Kapelner & Chandler, 2010; Oppenheimer, Meyvis, & Davidenko, 2009) to filter participants' qualifications, including:

1. **Java basic knowledge:** we asked participants a fill-in-the-blank question. Participants were not expected to answer this question completely correctly. However, they should at least pin point one correct Class from Java Standard Library.

   "What is Wrapper class? Please provide an example from *java.lang* package".

2. **"Kapcha" technique:** We set a delay for each question, so that participants were not able to go through all the questions rapidly. We assumed that a serious and consistent participant should always give the same response (i.e. the content of an alternative) to duplicated questions. So two more filters utilizing duplicated questions were to rule out these unserious responses.
3. **Duplication filter:** In close duplication filter, a duplicated question of the nth question would appear right after the nth question or the (n+1)th question with different order of alternatives. Participants were explicitly to be asked to select the same alternative in the new order.

```
Based upon your prior knowledge, arrange the sorting
techniques in increasing order of their time complex-
ities (average case scenario).
```

-Bubble Sort (B)
-Insertion Sort (I)
-Merge Sort (M)
-Quick Sort (Q)

(a) B< = I<M< = Q

(b) I = M = B = Q

(c) M = Q<I = B

(d) B<M<Q<I

We repeat the same question as above, with the same alternative in new order.

(a) B<M<Q<I

(b) Q = M<B = I

(c) B< = I<M< = Q

Upon completing qualification-filtering survey, participants will be directed to the task survey to begin indexing concepts to programming questions. Every qualified turker is asked to index Higher level concept(s) and Fine-grained concept(s) for 5 different questions. Each of participants receives $0.50 upon finishing the experiment. If participants who failed to do so, they would be redirected to a page telling them their non-qualifications. Qualified participants would be redirected to the page showing the consent form and start the main indexing task survey after filling out the consent form. The study was hosted on MTurk for 2.5 weeks. There are 406 turkers attempted the qualifier survey, 149 passed the qualification filters, and 103 of them completed the indexing tasks. Each of exam question in the corpus is indexed by at least 2 to 5 experts from the crowd.

4.2. Evaluation metrics

We assumed an effective algorithm would be able to index more relevant concepts or at least as many as experts did. In addition, a good quality algorithm should identify more key concepts as well as identify peripheral concepts. To verify our assumptions, we considered the following measures: **Concept Coverage, Concept Diversity, Concept Distinctiveness & Emphasis and Coherence.**

[2] https://www.qualtrics.com/.
[3] https://www.mturk.com/mturk/welcome.

4.2.1. Concept coverage

We defined the number of concepts indexed for each question as the concept coverage statistics (Eq. (1)).

$$coverage = \frac{indexed\ concept}{question} \tag{1}$$

4.2.2. Concept diversity

To gauge concept-indexing quality on exam questions based on the semantic topic facets generated by the algorithm, we assume that a meaningful question not only has to achieve high concept indexing coverage, but also has to encompass as many essential concepts as possible. We calculated *Shannon entropy* to estimate the breadth of semantics in gauging the indexed concept diversity of a given question (Eq (2)). We used *Shannon entropy* among several other diversity measures, such as mean of Euclidean distance, or standard deviations to measure two entities'separation, Gini coefficient to measure the disparity, and entropy, the state of the art measure of variety (Harrison & Klein, 2007) due to the same measure has been evaluated in similar context (Hsiao & Naveed, 2015; Momeni, Tao, Haslhofer, & Houben, 2013). Thus, concept diversity is defined as the following where q is the indexed concept of an exam question, *c* is a concept, and *n* is the number of concepts.

$$Entropy(\widehat{q}) = -\sum_{j=1}^{n} p(c_{i,j}) \log_e p(c_{i,j}) \tag{2}$$

4.2.3. Concept coherence

Since a question is being indexed with more distinct concepts does not always mean the better quality is achieved. For instance, if the concept terms are fine-grained concepts that are derived from the same hierarchy, they essentially represent the same ontological concepts. Thus, we hypothesized that an effective exam parser will identify a representative set of concepts to discern crucial semantics. Here is a concrete example with two sets of concept terms about a same question to illustrate the definition of good quality concept sets:

Question:"*Write an enhanced for loop that iterates through your ArrayList of decimal numbers and displays their sum to the console.*

```
double sum = 0;
for(Double e: aList)
   {
      sum = sum + e;
   }
   System.out.println(sum);"
```

4.3. Concept

C_A: *{SimpleVarible, DoubleDataType, DoubleValue, VariableInitializationStatement, ForEachStatement, ArrayList, WrapperClass, ArithmeticExpression, ArithmeticAssignmentExpression, AddExpression, JavaStandardLibraryClass, JavaStandardLibraryObject, JavaStandardLibraryMethod}*

C_B: *{SimpleVarible, DoubleDataType, DoubleValue, VariableInitializationStatement, ForEachStatement, ArrayList, WrapperClass, ObjectVariable, ArithmeticExpression, ArithmeticAssignmentExpression, AutoBoxing, JavaStandardLibraryClass, JavaStandardLibraryMethod}*

Both C_A and C_B had the same concept coverage and diversity. Both represented a good grasp of concepts in the given question. Yet, C_B was comparably a better quality set to depict the question than C_A. The underlined concepts highlighted the differences between C_A & C_B, where the distinct concepts *AddExpression* and *JavaStandardLibraryObject* did not contain any transitive relations with other concepts in the domain ontology. On the other hand, *ObjectVariable* and *AutoBoxing* in C_B appeared to be *partOf, hasPart* or *relatedTo* some other concepts in the domain (i.e. *ObjectVariable* is *relatedTo ObjectReference*, it is also *partOf MethodInvocation*). Thus, C_B presented higher semantic interrelations, which can be used to indicate better quality than simple distinct concept frequency counts. In order to capture the concept co-occurrence for a given question within the corpus, we adopted UMass topic coherence measure (Eq. (3)). Where *Q(x, y)* counts the number of questions containing concepts *x* and *y* and *Q(x)* counts the number of questions containing *x* in our exams corpus. It is a common measure for assessing language model based topic coherence by calculating the pairwise mutual information (Hsiao & Naveed, 2015; Mimno, Wallach, Talley, Leenders, & McCallum, 2011; Stevens, Kegelmeyer, Andrzejewski, & Buttler, 2012).

$$\mathrm{score_{UMass}}(c_i, c_j, \varepsilon) = log \frac{Q(c_i, c_j) + \varepsilon}{Q(c_i)} \tag{3}$$

4.3.1. Concept distinctiveness and emphasis

Based on information theory, we define *concept distinctiveness* as how informative the specific concept is for determining the indexed concepts, versus a randomly selected concept ***c'*** from ontology (Eq. (4)). For example, if a concept ***c*** occurs in all questions, identifying the concept tells us little about the exam's concept mixture; thus the concept would receive a low distinctiveness score. Therefore, we computed Kullback-Leibler divergence (Kullback & Leibler, 1951), for a given concept **c**, the conditional probability ***P(T|c)*** (the likelihood that observed concept **c** was generated by latent topic ***T***) and the marginal probability ***P(T)*** (the likelihood that any randomly selected concept ***c'*** was generated by topic ***T***). By calculating the product of *distinctiveness* and teachers' configuration weights ***P*(c)**, we will obtain the concept *emphasis* (Eq. (5)). Such measure has been successfully deployed in detecting topic model based topical conformity in detecting useful comments (Mimno et al., 2011), topic saliency to explore large text corpora themes (Chuang, Manning, & Heer., 2012) and topic sensitivity in online question answer communities (Zhou, Basu, Mao, & Platt., 2012).

$$distinctiveness(c) = \sum_{T} P(T|c) \log \frac{P(T|c)}{(T)} \tag{4}$$

$$emphasis(c) = P(c) \times distinctiveness(c) \tag{5}$$

5. Evaluation results

To evaluate the proposed intelligent indexing method for programming problems, we performed (1) algorithmic evaluation to measure *ExamParser* effectiveness in terms of indexed concepts coverage and quality; (2) content evaluation to examine the semantic approach impacts on exam question content; and (3) subjective evaluation by collecting interview feedback from teachers to understand their concerns and perceived usefulness.

5.1. ExamParser effectiveness

5.1.1. Baselines: experts' differences

We found that, not surprisingly, teachers indexed significantly fewer concepts than the crowd did (lower coverage), which supported the claim that teachers tend to point the *key* concepts for exam questions. However, we found that even teachers chose fewer concepts, they were able to achieve significant higher diversity among those selected concepts than the crowd experts did, ($p < 0.001$) per Table 1. The experts' variance suggests that the crowd may have indexed several concepts they are essentially close meaning to one another, while teachers tend to index uniqueness of the question. The result implies that experts from the crowd may have attempted to index as more comprehensively as possible. As we discussed earlier in section 4.2.2, the more doesn't always mean the better. The best scenario of a indexed question, it must consist of important concepts as well as secondary ones. Thus, we have to further look at the automatic indexing effectiveness.

5.1.2. ExamParser indexed more concepts with higher diversity

ExamParser indexed 5.36 concepts per question on average, it was significantly higher than the teachers, $t(75) = 9.465, p < 0.001$ (Table 2). It showed that the *ExamParser* could extract more concepts in general. *ExamParser* also showed no differences with experts from the crowd. This demonstrates that *ExamParser* could cover as comprehensive as the experts from the crowd. However, as we discussed earlier in 4.2.2, more concepts did not always represent a better coverage, if there were a lot repetitions or shallow concepts, the quantity growth didn't add more value in the semantic level. Thus, we measured the indexed concept diversity to gauge the comprehensiveness of the concept indexing. We found that *ExamParser* outperformed teachers in achieving significant higher concept entropy, $t(75) = 3.433$, $p < 0.001$ (Table 2). In addition, *ExamParser* also surpassed experts from the crowd, $t_{HLC}(75) = 7.45, p < 0.001, t_{FGC}(75) = 4.31, p < 0.001$. It demonstrated that the automatic concept indexing method not only extracted more concepts, but also extracted more diverse concepts. This is an encouraging note to traditional paper-based exams courses owing to the fact that it has always been challenging for teachers to spend a lot of class time to discuss every single detail of each exam question. It is common for teachers to focus on *selected* concepts (hopefully all *key* concepts) instead of all concepts. Therefore, automatic indexing method addresses more comprehensive concepts can systematically provide such *concept & question associations* as semantic feedback for exam questions. In addition, algorithmic indexing method can also be applied in parallel with human expert's indexing to complement each other as what teachers may have missed to mention in class. By tracing all these detail concept & question associations via learning analytics supplies additional learning opportunity allowing students to engage with realistic and persistent learning performance tracking.

Table 1
Baselines comparisons.

Average	Baseline I (Teacher)	Baseline II (MTurk)	
		High-Level Concept	Fine-Grained Concept
Coverage	2.35 ± 0.93	6.33 ± 3.05	10.93 ± 7.69
Diversity	3.67 ± 0.44	1.95 ± 1.41	2.01 ± 1.85

Table 2
Concept coverage & diversity.

Average	Teacher	*ExamParser*
Coverage	2.35 ± 0.93	5.36 ± 0.17
Diversity	3.67 ± 0.44	4.03 ± 0.86

5.1.3. ExamParser made important concepts salient

We have proven that *ExamParser* can index detail concepts of exam question semantics. However, did the detail concepts include essential concepts, in terms of important concepts being recognized from the exam questions, or just a bunch of shallow concepts? To verify the indexed concept quality, we further examined the extracted concept coherence and the concept distinctiveness to measure the indexed concept quality. We found that concepts indexed by *ExamParser* were significantly more coherent within the corpus than they were indexed by experts, $t(75) = 11.732, p < 0.001$ (Table 3). It was understandable that teacher experts tended to pinpoint the key concepts of each question instead of listing all peripheral ones. Thus, not surprisingly, when we looked at the *distinctiveness* scores (how informative the indexed concepts are), we found that teachers actually achieved significantly higher compared to *ExamParser*, $t(66) = 8.694$, $p < 0.001$. However, there were no distinctiveness differences between experts from the crowd and *ExamParser*, which means automatic indexing method could perform just as well as experts.

Moreover, EduAnalysis implemented the interactive visualization authoring interface for teachers to configure the importance of indexed concepts, thus preventing crucial concepts from being missed during exam parsing. Therefore, after configuring the relative importance weights for questions, we found that *emphasis* scores were found significant higher distinction from baseline, $t(66) = 12.529, p < 0.001$, where we assumed the distinctiveness of Baseline I and Baseline II (HLC) have already highlighted the emphasis. The authoring feature enabled teachers to highlight the essential concepts of each question, and compliment what algorithms might possibly miss or mis-index. The feature also empowered the learning analytics to be able to calculate and track partial credits based on the semantics, rather than individual grading rubrics provided by instructor.

5.2. Content influences

We consider the following aspects to assess analytics impacts on domain content: *Content Complexity & Content Knowledge Structure*.

Table 3
Indexing quality: concept coherence & concept distinctiveness & emphasis.

Average	Baseline I (Teacher)	Baseline II (MTurk)		*ExamParser*
		HLC	FGC	
Coherence	2.25 ± 1.84	16.93 ± 8.59	24.04 ± 14.48	12.91 ± 8.53
Distinctiveness	2.28 ± 0.79	1.74 ± 0.69	1.55 ± 0.49	1.69 ± 0.54
Emphasis	2.28 ± 0.79	1.74 ± 0.69	–	11.53 ± 6.06

5.2.1. *Content complexity*

According to CS1 course curriculum, depending on the exam foci topics, we split each exam by three levels of complexities, easy, moderate and complex. For instance, first exam usually covers topics from *variables*, *primitive data types, arithmetic operations, Strings*, *conditions* etc. These topics are usually considered relatively easy in the entire CS1 curriculum. However, in order to assess students' knowledge, an early CS1 test usually is devised with a mixture of difficulty levels questions. Thus, a question comprising of multiple topics was considered as a complex question in that exam. We have tabulated two interesting findings (Table 4). Firstly, baseline I group appeared to have no differences among complexity levels. It again supported the point that teacher experts tended to point out *key* concepts, instead of all concepts. Secondly, *ExamParser* indexed significantly more concepts in complex questions than the other two categories (Fig. 6). This result was very encouraging. More complicated questions were usually the ones that students made mistakes, which suggested more attention was demanded. However, as we discussed before, teachers may not necessarily have the class time to go through details on every single questions. Even if they did, such as mentioned *key* concepts of the tougher questions, the amount of feedback may not be sufficient. This where the *ExamParser* can make a difference by supplying more detail feedback.

5.2.2. *Knowledge structure: procedural vs. declarative knowledge*

In order to address the cognitive aspects of our approach's impact on learning content, we analyzed the indexed exam questions based on their knowledge types, procedural knowledge and declarative knowledge. A coarse-grained definition on procedural knowledge explains *one knows how to do something*; declarative knowledge approximately defines the knowledge about something. Thus, we identified the majority of the code writing questions were to test students' procedural knowledge, and most of the multiple choices questions were designed to assess declarative knowledge. However, there were a few exception cases did not follow such classification. For instance, in one of the code-writing questions, students were asked to write Java code to *"Instantiate an ArrayList that contains decimal numbers and assign it to an appropriate variable*". The question only involved syntactical tasks of the programming language, but excluded the application of syntax to perform further problem solving tasks. Thus, even though it was a code-writing question, it was classified as declarative question. Overall, we found 55% procedural questions and 45% declarative questions in the corpus. Based on the indexed concepts both by human experts and *ExamParser*, we found that, both types of questions had significant higher concepts indexed by *ExamParser* than the experts. This was consistent with 5.1.1, where *ExamParser* achieved higher coverage. What was interesting to note was that there were no significant differences between declarative and procedural knowledge types of questions, no matter who and how the questions were indexed. It showed the consistency among experts and the algorithm, which indicated *ExamParser's* stability. Although, we anticipated procedural type questions would have been indexed more concepts due to *knowing how to do* may inherently involve some declarative knowledge components in addition to apply them to solve problems. However, we did not find such pattern observed. Possible explanations could be declarative types of questions (i.e. multiple choices) tend to include a range of meaningful distractor choices to prevent from simple memorization tasks. It also explained why there were larger variances in declarative type of questions compared to procedural ones (Fig. 7 & Table 5).

Table 4
Average # indexed concept by content complexity.

Avg #concept	Baseline I	*ExamParser*
Easy	1.943 ± 0.121	4.400 ± 0.541
Moderate	2.318 ± 0.253	4.455 ± 0.443
Complex	2.316 ± 0.410	8.158 ± 0.763

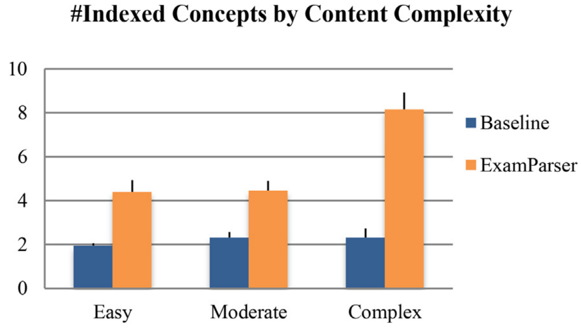


Fig. 6. Average indexed concepts per question by content complexity.

5.3. *Subjective feedback*

We conducted a structured interview with two programming course instructors. Both are currently using Blackboard as course management platform and both give lectures and paper-based exams. One teaches medium size of Java courses (20–50 students averagely) and one teaches large size of courses (>100 students averagely). We were mostly interested in finding out how do instructors analyze students' learning activities outside classrooms if any. Both instructors provide extra online learning materials (i.e. problem-solving resources or the sort) for students to perform self-assessments as non-mandatory resources for their courses. They encourage students to do more work through the selected online resources and provide partial credits for their academic performance as incentive.

We then allowed both instructors to explore EduAnalysis system and solicited feedback on the usefulness and potential threats of current implementation. They were instructed to test on the

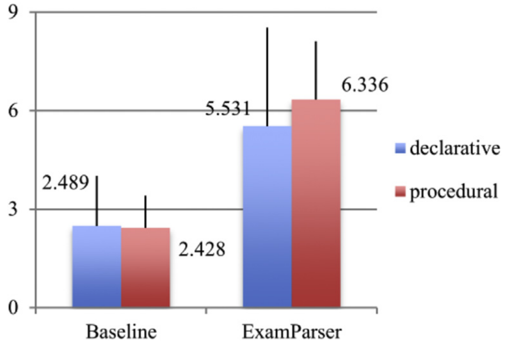


Fig. 7. Average indexed concepts per question by knowledge type.

Table 5
Concept coverage by knowledge type.

Concept/question	Baseline I	*ExamParser*
Declarative	2.489 ± 1.527	5.531 ± 3.000
Procedural	2.428 ± 0.986	6.336 ± 1.776

concept indexing procedure for different types of questions. They tried multiple choices and open-ended questions, and both agreed that the dynamic concept indexing provided them immediate feedback on producing more balanced exams. Both instructors reported that they found it convenient to perform one-click to upload and index exam concepts. They compared the experience with Blackboard evaluation feature, which requires them to configure each question one by one. Although the indexing authoring interface is available for every question, instructors considered it as flexible to assign designated emphasis to accommodate CS1/CS2 exams, or first/final exams. There were two major criticisms from both instructors: (1) they worried the auto-indexing precision may not be stable and result in them doing more configurations (which we proved in previous section, that the *ExamParser* appeared to be effective than experts' manual indexing); and (2) the usability was not conclusive at the moment, at least not until they adopt the tool for their courses. However, both instructors expressed the current semantic visual analytics was reasonably useful, and both indicated extreme interests in using it in their own classes in the future.

6. Discussions and conclusions

In this work, we designed and studied a semantic indexing method via visual analytics, EduAnalysis., which indexes paper-based programming problems to sets of concepts. It embeds intelligent concept indexing support to assist teachers in analyzing exam semanitc composition in detail. We collected the indexing ground truths of the targeted set from teachers and experts from the corwd. We evaluated the effectiveness of the indexing services, the indexing effects on content and investigated instructors' experiences and perceived usefulness on the system.

We found that current approach could extract significantly more and diverse concepts from exams, which enriched more semantic information. The findings supported our hypothesis that the intelligent indexing method and interactive visualization interfaces can facilitate paper-based programming content analysis. Such results unlock several opportunities to (1) make persistant traces of learning analytics in semantic level; (2) provide more personalized feedback for students that is normally difficult to achieve or afford in a traditional (large) classroom. In addition, we found that EduAnalysis empowered teachers to configure exam topical emphasis and the results of indexed concepts appeared to maintain coherence within exam. It suggested that the proposed *ExamParser* approach could potentially make it possible to assign partical credits by concepts. We also discovered that the *ExamParser* indexing effect was especially prevalent for complex content. The results complemented the cases when teachers could not afford a lot of class time, but were forced to discuss *key* concepts on the tougher problems on a returned exam. Moreover, we also found the automatic indexing method was consistent with teacher experts in indexing both procedural and declarative types of questions. Subjective evaluation revealed that dynamic concept indexing provided teachers immediate feedback on producing more balanced exams; teachers expressed strong interests in using EduAnalysis for their own classes. Overall, the semantic enriching approach for programming problems revealed systematic learning analytics from the paper exams.

In summary, we tested an intelligent semantic indexing for paper-based programming problems for orchestrating today's programming classes, by integrating physical classroom learning assessment (paper-based exams) to online visual learning analytics. Results indicated the automatic concept extraction from exams were promising and could be a potential technological solution to address a real world issue without tampering teachers' instruction pedagogy. There were a few limitations under current study setup, discussions were noted in the following section.

6.1. Limitations

Current exams selection was a sample of CS1 four exams from first author's home university, which is only a limited pool of exams. We should consider a wider range of exams and questions, such as textbook sample exams etc. There were a few evaluation limitations; such as teacher experts' Cohen Kappa indicated moderate agreement in our baseline I. As a result, the automatic *ExamParser* could potentially easily outperform experts. However, we argued that one of the reasons the inter-raters' agreement was low could be due to the nature of indexing challenges and the setup for experts to pick out concepts from a long list of ontology. In addition, teachers were used to identifying *key* concepts even though they were instructed to be as comprehensive as they could when indexing. Given that the teachers' ground truth was not perfectly satisfying, we did not measure indexing error rate at this moment.

6.2. Future work

In the near future, we need to address the teachers' concerns and to improve current design and evaluation. We plan to conduct field studies to collect larger scale of actual classroom usages and evaluate the semantic learning analytics impacts on students' learning. For instance, on a returned exam to student, besides receiving the grade marks, student will receive systematic semantic feedback based on what kinds of errors they made on the exams. We anticipate the enriching programming semantics approach will provide (1) individualized detail conceptual feedback, which normally can't be done especially in large class size; (2) analytics to keep persistent traces on students' conceptual growth; (3) opportunities for students to engage in reflection and self-monitor their own learning (foster metacognition development). Finally, nevertheless, we will integrate other learning activities for more comprehensive analysis. More exhaustive evaluation is required.

Acknowledgements

The system was implemented as part of Sesha Kumar Pandhalkudi Govindarajan master thesis. We thank Sesha's efforts in development and maintenance.

The second author was sponsored by MOST 105- 2410-H-110-019 - and Sayling Wen Cultural & Educational Foundation and Service Science Society of Taiwan.

References

Alomari, J., Hussain, M., Turki, S., & Masud, M. (2015). Well-formed semantic model for co-learning. *Computers in Human Behavior, 51*, 821–828.

Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010). *How learning works: Seven research-based principles for smart teaching*. John Wiley & Sons.

Amirkhani, H., & Rahmati, M. (2014). Agreement/disagreement based crowd labeling. *Applied Intelligence, 41*(1), 212–222.

Aroyo, L., & Dicheva, D. (2001). AIMS: Learning and teaching support for www-based education. *International Journal for Continuing Engineering Education and Life-Long Learning, 11*(1/2), 152–164.

Awasthi, P., & Hsiao, I. (2015). INSIGHT: A semantic visual analytics for programming discussion forums. In *1st international workshop on visual approaches to learning analytics in conjunction with Learning Analytics & Knowledge Conference*. Poughkeepsie, NY: Marist College.

Basu, S., Jacobs, C., & Vanderwende, L. (2013). Powergrading: A clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics, 1*, 391–402.

Bateman, S., Brooks, C., Mccalla, G., & Brusilovsky, P. (2007). Applying collaborative tagging to e-learning. In *Proceedings of the 16th international world wide web conference*.

Brusilovsky, P., Somyürek, S., Guerra, J., Hosseini, R., & Zadorozhny, V. (2015). The value of social: Comparing open student modeling and open social student modeling. In *International conference on user modeling, adaptation, and personalization* (pp. 44–55). Springer International Publishing.

Buhrmester, M., Kwang, T., & Gosling, S. D. (2011). Amazon's Mechanical Turk a new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science, 6*(1), 3–5.

Bull, S. (2004). Supporting learning with open learner models. In *4th hellenic conference on information and communication technologies in education,. Athens, Greece*.

Bull, S., & Kay, J. (2016). SMILI☺:: A framework for interfaces to learning data in open learner models, learning analytics and related fields. *International Journal of Artificial Intelligence in Education*, 1–39.

Chen, Z.-H., Chou, C.-Y., Deng, Y.-C., & Chan, T.-W. (2007). Active open learner models as animal Companions: Motivating children to learn through interacting with my-pet and our-pet. *International Journal of Artificial Intelligence in Education, 17*(2), 145–167.

Chuang, J., Manning, C. D., & Heer, J. (2012). Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the international working conference on advanced visual interfaces* (pp. 74–77).

De Liddo, A., Shum, S. B., Quinto, I., Bachler, M., & Cannavacciuolo, L. (2011). Discourse-centric learning analytics. In *Proceedings of the 1st international conference on learning analytics and knowledge* (pp. 23–33). ACM.

Demartini, G., Difallah, D. E., & Cudré-Mauroux, P. (2012). ZenCrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on world wide web* (pp. 469–478).

Dicheva, D., Dichev, C., Sun, Y., & Nao, S. (2004). Authoring topic maps-based digital course libraries. In *The workshop on applications of semantic web technologies for educational AH, in conjunction with AH'04* (pp. 23–26). The Netherlands: Eindhoven.

Dillenbourg, P. (2013). Design for classroom orchestration. *Computers & Education, 69*, 485–492.

Dimitrova, V., Self, J., & Brna, P. (2001). Applying interactive open learner models to learning technical terminology. In M. Bauer, P. J. Gmytrasiewicz, & J. Vassileva (Eds.), *8th international conference on user modeling, UM 2001* (Vol. 2109, pp. 148–157). Berlin: Springer-Verlag. Sonthofen.

Duval, E., Verbert, K., Klerkx, J., Wolpers, M., Pardo, A., Govaerts, S., et al. (2015). VISLA: Visual aspects of learning analytics. In *Proceedings of the fifth international conference on learning analytics and knowledge* (pp. 394–395). ACM.

Dyckhoff, A. L., Zielke, D., Bültmann, M., Chatti, M. A., & Schroeder, U. (2012). Design and implementation of a learning analytics toolkit for teachers. *Journal of Educational Technology & Society, 15*(3), 58–76.

Edwards, S. H., & Perez-Quinones, M. A. (2008). Web-CAT: Automatically grading programming assignments. In ACM SIGCSE. *Bulletin, 40*(3) (328–328). (ACM).

Epp, C. D., & Bull, S. (2015). Uncertainty representation in visualizations of learning analytics for learners: Current approaches and opportunities. *Learning Technologies, IEEE Transactions on, 8*(3), 242–260.

Franklin, M. J., Kossmann, D., Kraska, T., Ramesh, S., & Xin, R. (2011). CrowdDB: Answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD international conference on management of data* (pp. 61–72). ACM.

Harrison, D. A., & Klein, K. J. (2007). What's the difference? diversity constructs as separation, variety, or disparity in organizations. *Academy of Management Review, 32*(4), 1199–1228.

Heffernan, N., & Heffernan, C. (2014). The ASSISTments Ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education, 24*(4), 470–497.

Hosseini, R., & Brusilovsky, P. (2013). JavaParser: A fine-grained concept indexing tool for Java problems. In *AIEDCS workshop. Conference proceedings, Memphis, USA*.

Howe, J. (2008). *Crowdsourcing: Why the power of the crowd is driving the future of business*. Random House LLC.

Hsiao, I. (2016). Mobile grading paper-based programming Exams: Automatic semantic partial credit assignment approach. In *The eleventh european conference on technology enhanced learning*. Lyon, France: Springers.

Hsiao, I., & Awasthi, P. (2015). Topic facet modeling: Semantic visual analytics for online discussion forums. In *Proceedings of the Fifth international conference on learning analytics and knowledge - LAK '15* (pp. 231–235).

Hsiao, I., Bakalov, F., Brusilovsky, P., & König-Ries, B. (2013). Progressor: Social navigation support through open social student modeling. *New Review of Hypermedia and Multimedia, 19*(2), 112–131.

Hsiao, I., & Brusilovsky, P. (2012). Motivational social visualizations for personalized e-learning. In *7th european conference on technology enhanced education (ECTEL)*. Saarbrücken, Germany: Springer-Verlag.

Hsiao, I., & Naveed, F. (2015). Identifying learning-inductive content in programming discussion forums. In *The frontiers in education conference (FIE). El paso, Texas, USA*.

Hsiao, I., Sosnovsky, S., & Brusilovsky, P. (2010). 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.

Jovanovic, J., Gasevic, D., Brooks, C., Devedzic, V., Hatala, M., Eap, T., et al. (2007). Using semantic web technologies to analyze learning content. *IEEE Internet Computing, 11*(5), 45–53.

Kapelner, A., & Chandler, D. (2010). Preventing Satisficing in online surveys. In *CrowdConf*.

Kardan, A. A., Sani, M. F., & Modaberi, S. (2016). Implicit learner assessment based on semantic relevance of tags. *Computers in Human Behavior, 55*, 743–749.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics, 22*(1), 79–86.

Loboda, T., Guerra, J., Hosseini, R., & Brusilovsky, P. (2014). Mastery Grids: An open source social educational progress visualization. In C. Rensing, S. de Freitas, T. Ley, & P. Muñoz-Merino (Eds.), *Open learning and teaching in educational communities* (Vol. 8719, pp. 235–248). Springer International Publishing.

Marcus, A., Wu, E., Karger, D., Madden, S., & Miller, R. (2011). Human-powered sorts and joins. *VLDB Endow, 5*, 13–24.

Martinez-Maldonado, R. (2014). *Analysing, visualising and supporting collaborative learning using interactive tabletops*. University of Sydney (Thesis).

Martinez-Maldonado, R., Dimitriadis, Y., Martinez-Monés, A., Kay, J., & Yacef, K. (2013). Capturing and analyzing verbal and physical collaborative learning interactions at an enriched interactive tabletop. *International Journal of Computer-Supported Collaborative Learning, 8*(4), 455–485.

Mazza, R., & Dimitrova, V. (2007). CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. *International Journal of Human-Computer Studies, 65*(2), 125–139.

Medelyan, O., & Witten, I. H. (2006). Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEEE-CS joint conference on digital libraries* (pp. 296–297).

Mimno, D., Wallach, H. M., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the conference on empirical methods in natural language processing*. Edinburgh, United Kingdom: Association for Computational Linguistics.

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.

Mohler, M., & Mihalcea, R. (2009). Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th conference of the european chapter of the association for computational linguistics* (pp. 567–575).

Momeni, E., Tao, K., Haslhofer, B., & Houben, G.-J. (2013). Identification of useful user comments in social media: A case study on flickr commons. In *Proceedings of the 13th ACM/IEEE-CS joint conference on digital libraries*. Indianapolis, Indiana, USA: ACM.

Oppenheimer, D. M., Meyvis, T., & Davidenko, N. (2009). Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of Experimental Social Psychology, 45*(4), 867–872.

Paolacci, G., Chandler, J., & Ipeirotis, P. G. (2010). Running experiments on amazon mechanical turk. *Judgment and Decision Making, 5*(5), 411–419.

Paquette, G. (2007). An ontology and a software framework for competency modeling and management. *Educational Technology & Society, 10*(3), 1–21.

Park, H., Pang, R., Parameswaran, A. G., Garcia-Molina, H., Polyzotis, N., & Widom, J. (2012). Deco: A system for declarative crowdsourcing. *Proceedings of the VLDB endowment, 5*(12), 1990–1993.

Rolling, L. (1981). Indexing consistency, quality and efficiency. *Information Processing & Management, 17*, 69–76.

Roschelle, J., Dimitriadis, Y., & Hoppe, U. (2013). Classroom orchestration: Synthesis. *Computers & Education, 69*, 523–526.

Roschelle, J., Penuel, W. R., & Abrahamson, L. (2004). Classroom response and communication systems: Research review and theory. In *Annual meeting of the American educational research association (AERA). San Diego, CA*.

Sharples, M. (2013). Shared orchestration within and beyond the classroom. *Computers & Education, 69*, 504–506.

Slotta, J. D., Tissenbaum, M., & Lui, M. (2013). Orchestrating of complex inquiry: Three roles for learning analytics in a smart classroom infrastructure. In *Proceedings of the third international conference on learning analytics and knowledge* (pp. 270–274). ACM.

Sosnovsky, S., Hsiao, I. H., & Brusilovsky, P. (2012). Adaptation "in the wild": Ontology-based personalization of open-corpus learning material. In A. Ravenscroft, S. Lindstaedt, C. Kloos, & D. Hernández-Leo (Eds.), *21st century learning for 21st century skills* (Vol. 7563, pp. 425–431). Springer Berlin Heidelberg.

Stevens, K., Kegelmeyer, P., Andrzejewski, D., & Buttler, D. (2012). Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 952–961). Association for Computational Linguistics.

Surowiecki, J. (2004). *The wisdom of Crowds: Why the many are smarter than the few and how collective wisdom shapes business* (Economies, Societies and Nations).

Tiropanis, T., Davis, H., Millard, D., & Weal, M. (2009). Semantic technologies for learning and teaching in the Web 2.0 era. *IEEE Intelligent Systems, 24*(6), 49–53.

Vatrapu, R., Teplovs, C., Fujita, N., & Bull, S. (2011). Towards visual analytics for teachers' dynamic diagnostic pedagogical decision-making. In *Proceedings of the 1st international conference on learning analytics and knowledge* (pp. 93–98). ACM.

Verbert, K., Duval, E., Klerkx, J., Govaerts, S., & Santos, J. L. (2013). *Learning analytics dashboard applications*. American Behavioral Scientist.

Zapata-Rivera, J. D., & Greer, J. E. (2000). Inspecting and visualizing distributed Bayesian student models. In *International conference on intelligent tutoring systems* (pp. 544–553). Springer Berlin Heidelberg.

Zhou, D., Basu, S., Mao, Y., & Platt, J. C. (2012). Learning from the wisdom of crowds by minimax entropy. In *Advances in neural information processing systems* (pp. 2195–2203).