SCALABLE TEACHING AND LEARNING VIA INTELLIGENT USER INTERFACES

by

Xiangmin Fan

Bachelor of Science, Shandong University, 2011

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH

KENNETH P. DIETRICH SCHOOL OF ARTS AND SCIENCE

This dissertation was presented

by

Xiangmin Fan

It was defended on

January 18, 2017

and approved by

Dr. Jingtao Wang, Department of Computer Science, University of Pittsburgh

Dr. Milos Hauskrecht, Department of Computer Science, University of Pittsburgh

Dr. Diane Litman, Department of Computer Science, University of Pittsburgh

Dr. Muhsin Menekse, School of Engineering Education, Purdue University

Dissertation Advisor: Dr. Jingtao Wang, Department of Computer Science, University of

Pittsburgh

iii

SCALABLE TEACHING AND LEARNING VIA INTELLIGENT USER INTERFACES

Xiangmin Fan, PhD

University of Pittsburgh, 2017

The increasing demand for higher education and the educational budget cuts lead to large class sizes. *Learning at scale* is also the norm in Massive Open Online Courses (MOOCs). While it seems cost-effective, the massive scale of class challenges the adoption of proven pedagogical approaches and practices that work well in small classes, especially those that emphasize interactivity, active learning, and personalized learning. As a result, the standard teaching approach in today's large classes is still lectured-based and teacher-centric, with limited active learning activities, and with relatively low teaching and learning effectiveness.

This dissertation explores the usage of Intelligent User Interfaces (IUIs) to facilitate the efficient and effective adoption of the tried-and-true pedagogies *at scale*. The first system is MindMiner, an instructor-side data exploration and visualization system for peer review understanding. MindMiner helps instructors externalize and quantify their subjective domain knowledge, interactively make sense of student peer review data, and improve data exploration efficiency via distance metric learning. MindMiner also helps instructors generate customized feedback to students at scale.

We then present BayesHeart, a probabilistic approach for implicit heart rate monitoring on smartphones. When integrated with MOOC mobile clients, BayesHeart can capture learners' heart rates implicitly when they watch videos. Such information is the foundation of learner attention/affect modeling, which enables a *'sensorless'* and *scalable* feedback channel from students to instructors.

We then present CourseMIRROR, an intelligent mobile system integrated with Natural Language Processing (NLP) techniques that enables scalable reflection prompts in large classrooms. CourseMIRROR 1) automatically reminds and collects students' in-situ written reflections after each lecture; 2) continuously monitors the quality of a student's reflection at composition time and generates helpful feedback to scaffold reflection writing; 3) summarizes the reflections and presents the most significant ones to both instructors and students.

Last, we present ToneWars, an educational game connecting Chinese as a Second Language (CSL) learners with native speakers via collaborative mobile gameplay. We present a scalable approach to enable authentic competition and skill comparison with native speakers by modeling their interaction patterns and language skills asynchronously. We also prove the effectiveness of such modeling in a longitudinal study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

PREFACE

I wish to express my sincere appreciation to those who have supported me during my Ph.D. study.

First of all, I am forever indebted to my Ph.D. supervisor, Dr. Jingtao Wang, for his patient and valuable guidance, advice, encouragement, and continuous support throughout these years. Dr. Wang first inspired my love of doing research, and gave me hands-on advice on every detail in a research project at my beginning stage. He always gave me constructive and insightful suggestions for whatever we discussed about. He helped me build my self-confidence in conducting rigorous and high-quality research to solve challenging real-world problems, which would benefit my whole life. Dr. Wang also helped me greatly in my career planning and job search. He is my role model for a researcher, mentor and teacher. I am proud to be his student and am very grateful for what he has done for me.

My sincere gratitude is reserved for my Ph.D. committee members, Dr. Milos Hauskrecht, Dr. Diane Litman, and Dr. Muhsin Menekse. They provided invaluable insights and constructive suggestions for my research work. They also provided helpful advice and support for my job search. Dr. Litman and Dr. Menekse were also my project mentors and collaborators— it was unforgettable and nice experience working together with them. I would also like to thank Dr. Alexandros Labrinidis for serving as my temporary advisor at Pitt and for the valuable guidance and advice he gave me in the past few years.

1.0 INTRODUCTION

"Education is the manifestation of the perfection already in man."

—*Swami Vivekananda*

Education is not only "a light that shows the mankind the right direction to surge", but also "an engine for the growth and progress of any society" [46]. People's demand of higher education is constantly rising. In the United States, the total undergraduate enrollment in degree-granting postsecondary institutions increased by 46% between 1990 and 2013, from 12.0 million students to 17.5 million students [69]. At the same time, state funding cuts, especially since the start of the recession, has led to major consequences for public colleges and universities, such as higher tuition levels and teacher layoffs [83]. As a result, increasing the class sizes has become a natural way to address the conflicting issues. Large class sizes seem well-suited for transmitting significant amount of information to a large number of people at low cost. Besides the United States, large class size is also becoming a worldwide phenomenon: Mulryan-Kyne found that at the undergraduate level large classes of between 300 and 1000 and even more are common in a number of countries [124]. *Learning at scale* is not only the trend in traditional classrooms, but also the norm in Massive Open Online Courses (MOOCs), which has experienced a rapid growth in recent years. By examining public data from 279 courses, Jordan [86] found the average course enrolled ~43,000 students.

One of the assumed benefits of large classes is that they are *scalable*, but this perceived scalability rests on one key assumption—the *quality* of teaching and learning in large classes is comparable to that of smaller classes. However, in practice, the massive scale of class could introduce challenges to both instructors and students and can consequently lead to less effective teaching and learning. It is difficult for instructors to gain an adequate understanding of students' needs (e.g., difficulties, confusions, etc.) and cater to these individual needs in a crowded class. More importantly, large class sizes challenge the adoption of proven pedagogical approaches and practices that work well in small classes, especially those that emphasize *interactivity*, *active learning* [118], and *personalized learning*. As a result, the conventional teaching method in today's large class teaching is still lectured-based and teacher-centric, with limited active learning activities, and with relatively low teaching and learning effectiveness. For example, a study shows that an adult learner can focus in a lecture for no more than 15 to 20 minutes while sitting passively in a lecture hall [119]. Students are able to recall only 25% of the learning material after 3 hours of a one-way lecture [116].

This dissertation explores the usage of *Intelligent User Interfaces* (IUIs) to facilitate the efficient and effective adoption of the tried-and-true pedagogies *at scale* (Figure 1). Intelligent User Interfaces are "human-machine interfaces that aim to improve the efficiency, effectiveness, and naturalness of human-machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse, and media (e.g., graphics, natural language, gesture)" [114]. Generally, an IUI could understand the user's needs and personalize or guide the interaction via modeling the domain knowledge and/or the user. Via proper understanding and modeling of both instructors and students, our approaches engage both of them more actively in the learning process in a scalable and efficient manner. Our approaches are also holistic—being inspired by proven

pedagogies, solving practical challenges, and being tested in real-world scenarios. Compared with using fully automatic Intelligent Tutoring Systems (ITS) to *replace* human instructors, our approaches *facilitate* and *assist* instructors without completely overhauling the current education ecosystem.
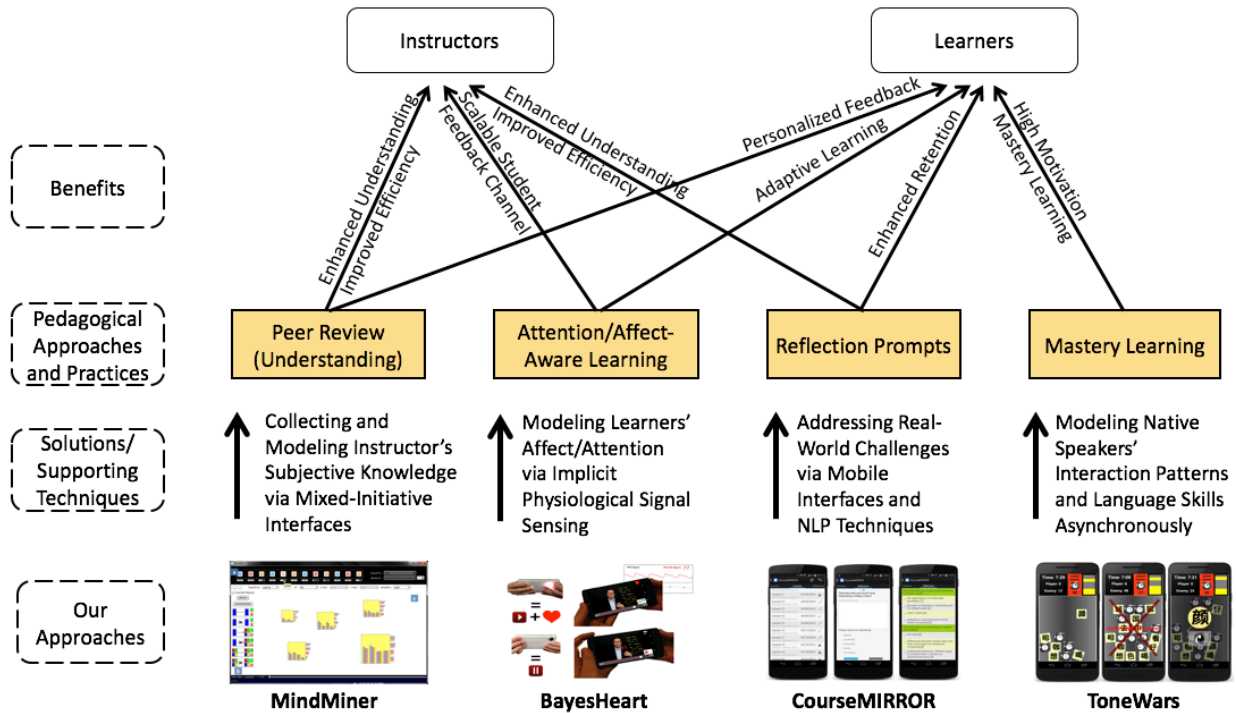
Figure 1. Major components (i.e. MindMiner, BayesHeart, CourseMIRROR, ToneWars) included in this dissertation.

1.1   DISSERTATION OUTLINE

The rest of the thesis is organized as follows:

Chapter 2 presents the recent innovations that scale and improve teaching and learning in large classrooms, as well as in MOOCs.

3

Chapter 3 presents **MindMiner** [56, 57]: an interactive data exploration and visualization system for instructors to understand student *peer review* data and generate customized feedback in a scalable manner. MindMiner collects and quantifies instructors' subjective knowledge on entity similarity via *mixed-initiative interfaces* and novel *machine learning algorithms*. MindMiner then uses such knowledge to group students with similar writing styles or writing problems together into clusters to improve data exploration efficiency. Such relevant clustering results generated by MindMiner could help instructors get a clear picture of the overall performance, find the underlying patterns, target students with similar writing problems thus give them customized feedback more efficiently. The content of this chapter can be found in the published papers [56] and [57].

Chapter 4 presents **BayesHeart** [55]: a commodity-camera-based photoplethysmography (PPG) sensing and probabilistic-based heart rate monitoring algorithm on unmodified smartphones. BayesHeart uses an adaptive hidden Markov model, requiring no user-specific training. When integrated with MOOC mobile client applications, BayesHeart can capture and collect learners' heart rates implicitly when they watch lecture videos. Such information is the foundation of learner attention/affect modeling, which enables a *'sensorless'* and *scalable* feedback channel from students to instructors. We released the source code of BayesHeart under BSD license at http://mips.lrdc.pitt.edu/bayesheart. The content of this chapter can be found in the published paper [55].

Chapter 5 presents **CourseMIRROR** [58, 59, 109]: a mobile learning system that uses natural language processing (NLP) techniques to enhance large classroom instructor-student interactions via streamlined and scaffolded *reflection prompts*. CourseMIRROR can 1) automatically remind and collect students' in-situ written reflections after each lecture; 2)

continuously monitor the quality of a student's reflection at composition time and generate helpful feedback to scaffold reflection writing; 3) summarize the reflections and present the most significant ones to both instructors and students. CourseMIRROR is freely available for classroom usage at: http://www.coursemirror.com. The content of this chapter can be found in the published papers [58], [59], and [109].

Chapter 6 presents **ToneWars** [60]: an educational game connecting Chinese as a Second Language (CSL) learners with native speakers via mobile gameplay. CSL Learners can practice tone recall, perception and production by competing with native speakers in ToneWars. We present a scalable approach to enable *mastery learning* by modeling both the *interaction patterns* and *fine-grained language skills* of native speakers asynchronously, so learners are able to play and practice at any time, regardless of native speaker availability. In our approach, native speakers serve as both a *benchmark* for language mastery and a *motivator* for language learning. The content of this chapter can be found in the published paper [60].

We conclude in Chapter 7 with a summary of major contributions and future work directions.

2.0   RELATED WORK

The unique challenges of teaching and learning in large classes, together with the development of technology, have prompted many to think about innovations to achieve high-quality education in both traditional large classrooms and MOOCs in a scalable manner. These approaches have show their feasibility and efficacy in previous studies and real-world deployments. In this section, we explore and discuss three research directions with different goals, target users, and potential benefits. Our intention here is to be exploratory and selective, rather than exhaustive, and to provide a series of snapshots of technologies that address the scalability issue in large classes.

We first explore and discuss the technologies that scale and promote proven pedagogies (e.g., emphasizing *interactivity*, *active learning*, and *personalized learning*) in large classes (Table 1). Such technologies can make it easier and more efficient to adopt the tried-and-true pedagogies in large classes and can benefit both instructors and students. The second category includes technologies that assist and facilitate human instructors to improve their work efficiency when dealing with a large number of students (Table 2). Such technologies can help them gain a better understanding of students' performances and needs and generate timely and personalized feedback in a scalable manner. Lastly, we discuss the recent development in intelligent tutors which aim to replace human instructors and to achieve simulated one-on-one instruction at low cost.

Table 1. Technologies that scale and promote active learning strategies.

Approaches	Major methodologies	Benefits for instructors	Benefits for students	Limitations
Audience response systems (ARS) (a.k.a. "classroom polling systems", or "clickers")	Real-time question posing and response collection [19, 38, 22, 42, 92, 68]; Result compilation and visualization [19, 38, 22, 42, 92, 68]	Immediate feedback about student comprehension [19, 38, 22, 42, 92, 68]; Improved interactivity and teaching effectiveness [38, 22, 42, 92]	Increased active participation and engagement [22, 38, 42, 92]; Enhanced retention of learning material [38, 42]; Increased attendance [19]	Hardware requirements and cost issues [22, 42]; Adaptations required in course planning [38]; Decreased lecture coverage [22]
Reflection/feedback collection systems	Response collection (e.g., explicit polling [80], spatially anchoring on lecture slides [66]); Visual summary for instructors [66]	Timely feedback about how well the students understood the lecture [66, 80]; Insight for lecture improvement [66]	Enhanced retention [66]; Improved active learning experience [66]	High demand for instructors' time [80]
Web-based peer review systems	Streamlined and automated reviewing process [32, 99, 158, 143]; Feedback/review quality assurance (e.g., calibration [143], detailed rubrics [32, 158, 99], norm-setting [99])	Reduced workload [32, 99, 158]	Rapid and sufficient amount of feedback [99]; Enhanced critical thinking and assessment skills [32, 99, 158]; Enhanced feeling of community [158]	Student bias; Lack of expertise; Lack of instructor's feedback [143]
Online discussion systems	Synchronous [25, 35] or asynchronous [91, 155, 74]; Enhancing online discussion structure (e.g., role specification [74], supportive interfaces [136])	Feedback from student by monitoring the discussions [91, 155, 74]	Increased participation, engagement, reflection, the the social construction of knowledge [91, 155]; Overcome isolation [25, 91]	Propagation of misconceptions [91]; Limited instructor control [91, 155]; Low participation rate [35]

2.1 SCALING AND PROMOTING PROVEN PEDAGOGIES

The first category includes technology that can scale and promote the tried-and-true pedagogies in large classes. Such approaches engage students more actively in learning and improve the instructor-student and student-student interactions, by scaling and promoting in-class communication (e.g., through audience response systems [22, 38, 42, 68, 92]), end-of-lecture reflections [66, 80], as well as collaborative and peer learning (e.g., through peer-review systems [32, 99, 158, 143] and online discussion systems [25, 35, 74, 91, 155]). These approaches can benefit students by engaging them more actively, and can also benefit instructors by providing them with sufficient amount of feedback from students.

Audience response systems (ARS, a.k.a. "clickers) in classrooms enable real-time question posing and response collection. The use of ARS has been promoted for its ability to focus student attention, identify gaps in knowledge, and enhance student engagement [38, 42, 92]. They also help to generate lively debate and promote in-class interactivity [22]. Previous studies have illustrated its effectiveness in improving attendance (e.g., increased attendance by 20% when the clicker points were worth 10% of the course grade [19]), and grades (e.g., increased the number of A's by 4.7% and decreased the combined proportion of students earning D's and F's by 3.8% [23]). Despite these benefits, most studies of clicker use agree that there is usually a decrease in content coverage when time is spent on ARS activities [22, 38]. Therefore, adaptations are required in course planning in order to achieve efficient and effective adoption. Besides, the hardware requirements and cost issues could prevent the widespread adoption of such systems.

Researchers and education practitioners also leverage peer review systems to enable rapid feedback [99], enhance students' motivation and engagement [32], and improve their communication skills (e.g., giving constructive criticism [158]). Peer reviews may be

8

fundamentally limited in that student reviewers are novices instead of experts in their disciplines [32]. As a result, their feedback and evaluation could be inaccurate and less helpful relative to the feedback generated by instructors. Multiple approaches were proposed to address this limitation and help student reviewers generate more specific, constructive, and helpful reviews and critiques. By integrating a calibration step before the real peer review step, [143] observed significant improvements in both students' writing skills and review competency in a study involving nine instructors and 789 students. Researchers also found that students can generate higher quality reviews when they are provided with detailed rubrics [32, 158, 99]. Previous study also suggested that providing instant feedback regarding the presence of solutions to students could help them generate more comments with solutions in peer reviews [125].

Online discussion systems (supporting either synchronous [25, 35] discussion or asynchronous [91, 155, 74] discussion) are proposed to enhance student-student interactions and increase participation, engagement, reflection, the the social construction of knowledge [91, 155]. To further improve the content quality of online discussion, [74] proposed to specify roles (i.e. starter, wrapper, instructor, student) before discussion and observed significant quality improvements. Despite these potential benefits, researchers reported at least three limitations in these systems: 1) propagation of misconceptions [91]; 2) limited instructor control[91, 155]; 3) low participation rate [35].

2.2   ASSISTING AND FACILITATING HUMAN INSTRUCTORS

The second category includes approaches that facilitate human instructors in large classes and improves their work efficiency.

Table 2. Technologies that assist and facilitate human instructors.

Approaches	Major methodologies	Benefits for instructors	Benefits for students	Limitations
Automated/ computer-assisted grading	Algorithms (e.g., answer key [87], similarity score[121], clustering [153]); Interfaces (e.g., clustering-based visualization [17])	Improved grading efficiency [153, 17]; Enhanced scalability [17]	Personalized feedback [17]	Imperfect accuracy [87, 121, 153]
Instructor-side data analytics and visualization systems	Learner-related data (e.g., grades [133], peer-reviews [175], video watching logs [93]) analysis and visualization; Used in both traditional classrooms [175, 133] and online courses [93]	Improved data exploration efficiency [133, 161]; Better understanding of students' learning [175, 133, 161, 93]; Insight for improving teaching [93]	Indirect benefit	

Multiple automatic/computer-assisted grading algorithms [87, 121, 153] and interfaces [17] have been proposed to facilitate grading and provide timely feedback to students in large classrooms and MOOCs. To enable open-ended question grading at scale, [87] proposed to automatically grade against an answer which includes all possible student answers. [121] formulated automated grading as a similarity task in which a score is assigned based on the similarity between the answer and correct answer. However, the accuracy of fully automatic grading is not 100% accurate (e.g., 84% in [87], 92% in [121]). Therefore, [17, 153] proposed to leverage both machine and human, specifically, automatically find groupings and subgroupings of similar answers from a large set of answers to the same question, and let teachers apply their expertise to mark the groups. They observed significant speed increase (i.e. 67%) compared with pure manual grading in a study with 25 teachers [17].

Instructor-side data analytics and visualization systems help instructors explore student-related data (e.g., grades [133], peer-reviews [175], video watching logs [93]) more efficiently. [161] presented a design space regarding tracked data of learning dashboard (e.g., time spent, social interaction, document and tool use, exercise/quiz/exam results, etc.) and analyzed 15 existing dashboards in the design space. These systems can provide instructors with insight and a better understanding of their students at scale.

2.3   INTELLIGENT TUTORS

The third category focuses on leveraging intelligent tutors [5, 43, 84, 89, 169] to replace human instructors and to provide students with simulated one-on-one instruction with relatively low cost.

The intelligent tutors first need to understand and model learners in order to provide individualized guidance accordingly in the teaching/learning process. Modeling learners' cognitive and affective states [84, 89, 5, 169, 43] via sensor data (e.g., physiological signals such as heart rates [169, 43], facial expressions [7, 89], eye gazes [169], and other types of data such as mouse pressure [7]) is a trend for today's intelligent tutors. After the signal sensing step, these systems leverage machine learning algorithms to predict students' affect and cognition and make adaptations accordingly—in terms of both the adapted learning content [169, 43, 84, 89, 5] and the adapted instructional strategies (e.g., interventions [169, 43])—to the students. Today's intelligent tutors also integrate pedagogical strategies for adaptation, e.g., mastery learning [89], and macro-adaptation [5]. However, the requirement of dedicated sensors is the main limitation of such systems. The cost and availability issues of such devices can prevent the wide adoption of these approaches in real-world settings.

In terms of learning outcomes, Kurt VanLehn's recent overview of modern ITS found that there was no statistical difference in effect size between expert one-on-one human tutors and step-based ITS [160]. Some ITS can even be superior to classroom teacher in certain settings [97]. However, given the current level of technological ability, ITS systems cannot lead or participate in deep discussions or debates as effectively as they deliver information. Therefore, they cannot fully replace the human instructors when students are at higher levels of inquiry [102]. Moreover, developing the intelligent tutors can be expensive, which may prevent the wide adoption of such systems in the wild.

3.0 MINDMINER: AN INTERACTIVE DATA EXPLORATION AND VISUALIZATION SYSTEM FOR PEER REVIEW UNDEERSTANDING

"Teaching peers is one of the best ways to develop mastery."

— *Jeff Atwood*

3.1 BACKGROUND AND INTRODUCTION

Peer review is a widely used pedagogy for coaching writing in many domains [159]. In this process, individual students take two roles: one of writer and one of reviewer [32]. Peer review becomes an important component of writing classrooms because it encourages *active learning* [118], giving students the opportunity to become more deeply engaged with their writing, and with one another. Moreover, peer collaboration is more effective to detect students' misunderstanding and contradictions which are unlikely to be detected when students working alone [113]. Besides, peer review also offers scalability to writing classrooms by reducing instructors' workload so that they can spend more time on other aspects of teaching [139].

However, peer review may be fundamentally limited in that student peer reviewers are novices in their disciplines. Thus, their feedback and evaluation could be inaccurate relative to the feedback generated by an expert or instructor. Therefore, the benefits of peer review still depend on the instructor actively reviewing, understanding students' peer review data (e.g., rubric-based

grade, comment text, etc.) and providing professional, customized and timely feedback to students. At the same time, instructors find that peer review data is time consuming to read and almost impossible to interpret [175] when they are facing a large number of students— it imposes high cognitive workload both in understanding one student's paper by synthesizing all peer reviews received by that student and in discovering general patterns by comparing peer reviews across multiple students.

In this chapter, we propose MindMiner [56, 57] (Figure 2), an interactive data exploration and visualization system for instructors to understand peer review data and generate customized feedback in a scalable manner. MindMiner maintains the teacher's involvement while still allowing them to work with peer review data efficiently at scale. MindMiner employs data visualization at multiple levels of granularity, and uses clustering to group students with similar writing styles or writing problems together into clusters to improve data exploration efficiency. We hypothesize that relevant and accurate clustering could help instructors get a clear picture of the overall performance, find the underlying patterns, target students with similar writing problems thus give them customized feedback more efficiently.

Cluster analysis is widely used in exploratory data mining and is desirable in that it is unsupervised and can discover the underlying structure of data without *a priori* information. However, to get meaningful and relevant clustering results, clustering algorithms expect a quantitative, deterministic distance function to quantify the similarity between two entities. In most real world problems, such similarity measurements usually require subjective domain knowledge that can be hard for users to explain. For example, a human instructor may easily find that the writing styles of two students are very similar to each other by reviewing their writing samples.

However, such perceived similarities may not be reflected accurately in the distance measurement between two corresponding feature vectors (e.g., rubric-based grades).

To capture and collect instructors' subjective domain knowledge on student similarity measurement, which is essential for high-quality clustering, MindMiner leverages two techniques: *active polling with uncertainty* and *example based visual constraint creation*. *Active polling with uncertainty* enables users to specify their subjective opinion on the *global* importance of a feature (including the value "not sure"), which improves the accuracy, and speed of the clustering results. *Example based visual constraint creation* allows to directly express their *a priori* domain knowledge via six types of constraints on the data samples being visualized. The constraint management interface allows users to browse existing examples, investigate the impact of each constraint, and discover conflicting conditions. MindMiner also provides interface level support that uses *active learning* to provide optional hints as to which examples might be more helpful for clustering. We also report how inequalities are formulated based on the collected *a priori* knowledge and how the inequalities are used in a convex optimization process to extract the "mental model" of entity similarity from users in the form of the Mahalanobis distance metric. The *intelligence* of MindMiner involves collecting, modeling and using instructors' subjective domain knowledge on student similarity to improve their efficiency in peer review exploration tasks.

The content of this chapter can be found in the published papers [56] and [57].

Figure 2. The primary UI of MindMiner, showing 23 students in a college-level philosophy class grouped into five clusters based on their performance (accuracy, clarity, and insight) in four writing assignments using six example constraints specified by an instructor. MindMiner consists of three parts: (a) The *Active Polling Panel* allows users to optionally indicate the importance for each measurement. Each colored square box represents one feature (4 assignments x 3 features). The rectangular bars beneath show real-time updates of the corresponding "weights"; (b) The *Constraints Management Sidebar* displays example-based constraints collected; (c) The *Interactive Visualization Workspace* lets a user see detailed information about entities, create example-based constraints, split and combine groups, examine and refine clustering results and examine personalized groups.

3.2   MINDMINER IN ACTION

We present a scenario giving an overview of MindMiner. MindMiner was originally designed for computer assisted peer-review and grading scenarios, but can also be used for other interactive clustering tasks.

Alice is an instructor for a philosophy course with 23 students. There are four writing assignments, and the essays submitted by students are graded via three features (accuracy, clarity, and insight). The grading is done by herself, the TA, and "double-blind" peer-review by students. Alice feels it is tedious and time consuming to get a clear picture of the overall performance of the whole class. Alice also wants to identify students with similar writing problems so that she can provide customized feedback to them. Alice can use MindMiner to achieve a balance between workload and feedback accuracy.

Figure 3. Knowledge collection interfaces of MindMiner. a: Interface for *active polling with uncertainty*. b: Interface for *example-based constraints collection*.

After logging into MindMiner, Alice retrieves student performance data from a remote server. Alice believes that writing accuracy is the most important factor she cares about and clarity a close second. She is not sure about the importance of insight. Therefore, she uses the *Active Polling Panel* (Figure 3.a) to make a choice for each feature. She chooses "very important" for accuracy, "important" for clarity and "not sure" for insight.

Then Alice teaches MindMiner her subjective judgments on performance similarity of students by labeling some example constraints. Alice reviews detailed information of the students by mousing over the nodes. MindMiner automatically selects the most potentially informative

pairs and highlights the suggestions with dashed lines (Figure 3.b). She examines two students involved in a constraint suggestion. After judging that they performed similarly, she drags them together, which creates a must-link constraint between the two students, telling MindMiner that these students should be grouped together. A corresponding symbol for this constraint then appears in the *Constraints Management Sidebar* (Figure 2.b). She later creates a cannot-link between dissimilar students by right clicking and dragging from one to the other. Every time Alice adds a new constraint, the distance metric learning module runs a convex optimization algorithm to derive the optimized solution. The bars in the *Active Polling Panel* (Figure 2.a) show the updated weights of corresponding feature dimensions in real-time.

MindMiner also checks if there are any conflicts caused by new constraints. If so, it gives a warning by highlighting the corresponding constraints in the *Constraints Management Sidebar* using a red background. Alice checks the conflicting constraints and finds that one of the previous example constraints she created is not correct so she deletes it. Each constraint item in the *Constraints Management Sidebar* is double-linked with corresponding students via mouse hovering, so it is easy for Alice to diagnose the cause when a conflict is reported by MindMiner.

Alice clicks the "group" button located on the top of the *Constraints Sidebar* to see whether the examples provided by her are sufficient for grouping students together in a useful manner. MindMiner applies the updated distance metric using a k-means clustering algorithm, and then displays the resulting groups. Alice then checks the results and finds that the groups are not as good as she expected. She adds a few more constraints and then she checks "automatic regroup". In this mode, once there is a new constraint, MindMiner's learning algorithm executes and the system automatically regroups the students based on the most updated distance metric. Alice

continues this iterative process by adding new constraints, deleting existing constraints or adjusting importance levels of the features, until she gets satisfactory clustering results.

3.3   RELATED WORK

Previous efforts have been made by researchers to improve the quality of clustering using both algorithmic [37], [162, 173] and user interface [26], [48], [51] approaches. For example, various semi-supervised clustering algorithms have been proposed by researchers in the machine learning community, either by adapting a similarity measure via user specified constraints or by modifying the process of determining intermediate cluster centers. However, most existing work focuses on *theoretical feasibility*: they assume users can provide sufficient, unambiguous, and consistent information to facilitate clustering before the algorithms start.

Researchers in HCI and Information Visualization have also explored the use of interactive applications for guided clustering [48], [77], [129], [149]. Some interfaces rely on real time feedback of clustering results to help users choose proper features, samples, and the number of clusters to use. Other systems, such as IVC [48], attempt to provide mechanisms to collect users' *a priori* knowledge, such as which samples should be in the same group, and which should not. However, most existing interactive clustering systems focus on *conceptual demonstration* and do not address important elements for making such systems practical, such as how to browse, how to manage users' collected *a priori* knowledge, and how to achieve better clustering results with more representative constraint examples.

3.4   DESIGN OF MINDMINER

In the following sections, we discuss these parts in more detail, including the visualization design, the knowledge collection interfaces in MindMiner and the underlying mathematical modeling and the convex optimization algorithm for learning the distance metric respectively.

3.4.1   Visualization Design

We use interactive stacked bar charts in MindMiner to visualize clusters of data with multivariate features. Figure 4 illustrates an example of our design in which a student dataset is visualized. Each student, treated as an entity, is characterized by his/her performances in a writing course along different features, i.e. accuracy, clarity, and insight. These features are defined by the user, and are measured based on the peer-review scores of three writing assignments.

As shown in Figure 4.a, we use different background colors to illustrate different assignments, and use different foreground colors to represent the different features. A student's feature vector is represented as a bar chart (Figure 4.b) in which the sizes of the bars represent the corresponding review scores. Similarly, we represent a clustered group of students (Figure 4.c) by packing all of the students' review scores together into a stacked bar chart, categorized by assignments (Figure 4.d). We also represent the averaged student feature scores of each assignment as another grouped bar chart attached to the group. The position of the bar chart, i.e. left, right (default location, Figure 4.d), bottom, and top, can be customized by users. The resulting visualization shows the overall distribution of data while keeping individual details easily visible.
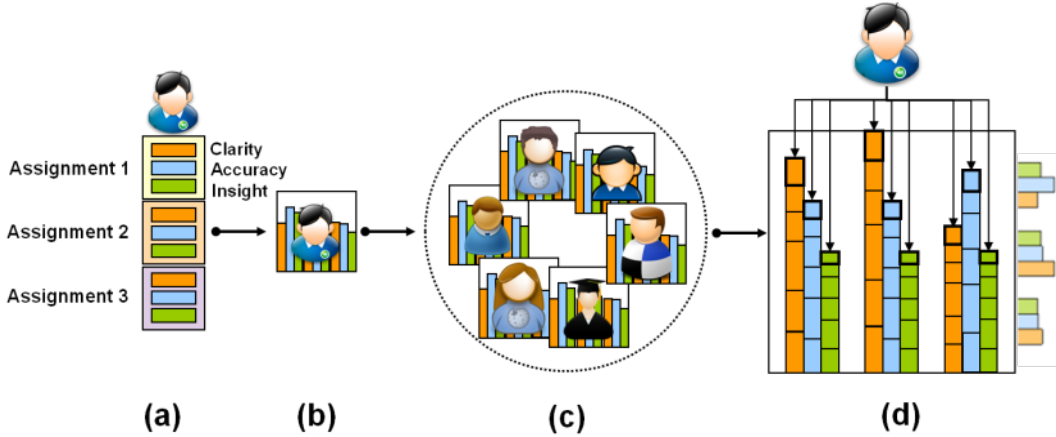
Figure 4. MindMiner visualization design. a) Feature vector of a student based on three writing assignments and three different features. b) Student barchart icon. c) A group of similar students. d) Stacked bar chart icon for a cluster of students.

3.4.2 Knowledge Collection Interfaces

MindMiner offers two novel knowledge collection techniques, *active polling with uncertainty* and *example-based constraints collection*, to make it easier for end-users to externalize their implicit mental models of entity similarity. We also introduce an active learning [36] heuristic to help users provide similarity examples that are more informative to the follow-up learning algorithms.

Active Polling with Uncertainty. MindMiner lets users specify their perceived importance of each feature via Active polling with uncertainty (Figure 3.a). Available choices are – "not important", "important", "very important" and "not sure". This step is optional and the default choice is "not sure". These choices correspond to different parameter search spaces in the convex optimization stage. As we illustrate later, expressing subjective certainty can reduce the number of examples needed in the next step and improve clustering quality.

Example-based Constraints Collection. MindMiner allows users to specify their knowledge on entity similarity via examples. This approach is supported by a psychology theory [144], which suggests that people represent categories through examples or prototypes.

Table 3. Symbols and descriptions of the six pairwise constraints supported by MindMiner. Collected constraints are shown in the *Constrains Management Sidebar* (Fig. 1.b).

Symbol	Name	Details
	Must-link	Lets user specify that two entities should be grouped together. Leads to a new entry in equation (2)
	Cannot-link	Lets user specify that two entities should not be grouped together. Leads to a new entry in equation (3).
	Must-belong	Lets user specify that one entity should be included in a specific group. Leads to multiple must-links, and added as multiple entries in equation (2)
	Cannot-belong	Lets user specify that one entity should not be included in a specific group. Leads to multiple cannot-links, and added as multiple entries in equation (3)
	Similar groups	Lets user specify that two existing groups should be put together. Leads to multiple must-links, and added as multiple entries in equation (2)
	Dissimilar groups	Lets user specify that no items in the two existing groups should be put into the other group. Leads to multiple cannot-links, and added as multiple entries in equation (3)

MindMiner supports six types of constraints (Table 3). All six constraints can be specified by users in the primary interface via mouse-based direct manipulation operations. Constraints created are shown in the *Constraint Management Sidebar* (Figure 2.b). This sidebar allows users to browse, remove, or check the impact of each constraint created. Conflicting constraints are also highlighted in red. These constraints are used in the inequality generation step later.

3.4.3   Mathematical Background

An entity in MindMiner is denoted by an n-dimensional feature vector. For example, entity s_i is represented by $(s_{i1}, s_{i2}, ..., s_{in},)$ in which n is the dimension in the feature space. The similarity measurement $d(s_i, s_j)$ between entity s_i and entity s_j is defined as:

$$d(s_i, s_j) = \sqrt{(s_i - s_j)W(s_i - s_j)^T} \tag{1}$$

Here W is an $n*n$ distance metric matrix. Letting $W=I$ leads to Euclidean distance. In MindMiner, we restrict W to be diagonal for efficiency concerns; the same framework can be used to learn a complete W with sufficient user examples. Determining each non-zero element in the diagonal W corresponds to learning a metric in which the different measurement features are given different "weights". Therefore, our goal here is to find W (weights vector) which best respects the information collected via the active polling process and interactive constraint creation process.

3.4.4   Constraint Conflict Detection

The information collected with *active polling with uncertainty* is used to define the lower and upper bound of the associated weight for each feature in the follow-up optimization process. The choice "Very important" corresponds to a weight of 1 (highest), "not important" corresponds to a weight of 0 (lowest), the weights of "important" features are set to be in a range of [0.6, 1] while "not sure" features are set to be within [0, 1]. In the end, we get a set of ranges for the weights of all features:

$$WeightBounds(WB) = \{[w_{1\_lb}, w_{1\_ub}], ... [w_{n\_lb}, w_{n\_ub}]\}$$

As shown in Table 3, depending on the constraint type, each constraint collected will be converted to one or multiple pairwise relationships and a Boolean flag. For must-link and cannot-link, the corresponding list only contains one pair, with a Boolean flag indicating the similarity relationship (true for similar and false for dissimilar) between the entities involved in the pair. For other types of constraints, they are first converted to multiple pairwise constraints such as must-links or cannot-links. Then these must-links or cannot-links are added to the pair-list of the corresponding constraint.

input : Existing constraints in C and the new added constraint c
output: All the conflicting constraint(s)

for *each exsiting constraint $C_i \in C$* **do**
    if *the similarity flags of c and C_i are different* **then**
        //conflicts may exist;
        iterate through the pairs lists of c and C_i to see if there is a
        common pair. If yes, mark C_i as a conflicting constraint.
    end
end

Algorithm 1. Constraint conflict detection.

By using this list based constraint representation, Algorithm 1 presents pseudo code to detect prospective conflicts in the constraints provided by end-users. If a constraint conflict is detected, corresponding constraints in the *Constraints Management Sidebar* (Figure 2.b) will turn red. Also, hovering over a conflicting constraint will highlight the remaining constraint(s) in conflict, as well as the corresponding entities and groups.

3.4.5 Active Learning Heuristic

Not all user-specified examples are equally helpful in improving the results from convex optimization. Some examples could be repetitive and would not justify the time spend by users to specify them or the extra computer-time added to the optimization process. Therefore, we adopted

24

concept of active learning, which allows MindMiner to identify and suggest *ambiguous entity relationships* that are most informative in improving the quality of distance metric learning. The informative entity pairs discovered via active learning are marked with dashed lines in the main interface.

3.4.6 Inequality Generation

We also keep two global sets: *S*, which is a set of pairs of entities to be "similar" and *D*, which is a set of pairs of entities to be "dissimilar". All the similar pairs are added to *S* while all the dissimilar pairs are added to *D* during the interactive constraint creation process.

A straightforward way of defining a criterion for the meaningful distance metric is to demand that pairs of entities in *S* have small squared distance between them (eq.2). However, this is trivially solved with *W=0* and is not informative. Our approach was primarily inspired by the method proposed by Xing et al. [173]. To avoid the trivial solution, we add a new inequality constraint (eq.3) to ensure it takes dissimilar entities apart. In this framework, we transform the problem of learning meaningful distance metrics to a convex optimization problem:

$$min_w \sum_{(s_i, s_j \in S)} d^2(s_i, s_j) \tag{2}$$

s.t.

$$\sum_{(s_i, s_j \in D)} d(s_i, s_j) \geq 1 \tag{3}$$

For each

$$w_k : w_k \geq 0 \; (1 \leq k \leq n) \tag{4}$$

Each sum item in eq.2 corresponds to a positive constraint collected, while each sum item in eq.3 corresponds to a negative constraint collected (Table 3).

It can be proven that the optimization problem defined by eq.2 – eq.4 is convex, and the distance metric W_{raw} can be solved by efficient, local-minima-free optimization algorithms.

Unfortunately, according to our early experiences on real world data, it is not desirable to use W_{raw} as the distance metric for the follow-up clustering tasks. According to our observations, when the number of constraints is very small, especially at the beginning of a task, convex optimization usually leads to a sparse distance metric where most values in the distance metric are close to zeros, i.e. only minimal features, e.g., 1 or 2 features, are taken into account in similarity measurement, implying a trivial solution that does not represent the real-world situation. We use an extra result regularization step and leverage the information collected in the *active polling with uncertainty* step to generate more meaningful distance metric that could be a better representation of a user's mental model.

3.4.7   Result Regularization

In order to make distance metrics respect both feature uncertainty information and the constraints collected by MindMiner, we regularize *Wraw* by using *Weight Bounds (WB)*. Detailed steps are described in Algorithm 2.

After finishing the result regularization step, we get a *W* that conforms to all the prior knowledge we collected from end-users. We apply *W* to the distance metric function and get the relevant distance metric. Then the distance metric *W* is used in k-means clustering algorithm to generate meaningful clusters.

```
input  : the raw weight $W_{raw_i}$ and the "lower bound" ($W_{i_{lb}}$) and "upper
         bound" ($W_{i_{ub}}$)
output: the regularized weight $W_i$ ($1 \leq i \leq n$)

Iterate through $W_{raw}$ and find the maximum and minimum values
$W_{raw-max}, W_{raw-min}$
for i from 1 to n do
   if $W_{i_{lb}} = W_{i_{ub}} = 1$ then
   |   set $W_i = 1$
   else if $W_{i_{lb}} = W_{i_{ub}} = 0$ then
   |   set $W_i = 0$
   else
   |   set $W_i = W_{i_{lb}} + (W_{i_{ub}} - W_{i_{lb}}) \cdot \frac{W_{raw_i} - W_{raw-min}}{W_{raw-max} - W_{raw-min}}$
end
```

Algorithm 2. Result Regularization.

3.5   EVALUATION

We conducted a 12-subject user study to understand the performance and usability of MindMiner. The data loaded in MindMiner in this study was anonymized real world data from a 23 student philosophy course in a local university with permission from the internal review board (IRB) and the instructor.

3.5.1   Experimental Design.

The study consisted of five parts:

Overview. We first gave participants a brief introduction and a live demo of MindMiner. We explained each task to them, and answered their questions. After the introduction, we let the participants explore the interface freely until they stated explicitly that they were ready to start the follow-up tasks.

Clustering and active learning. We used a within-subjects design in this session. There were two similar tasks: task 1 was clustering the students into four groups based on their performance in the first assignment; task 2 was the same as the previous task except that users were to only consider the "accuracy" features of the assignments. There were two conditions in this section: (A) providing constraint suggestions via active learning; (B) without active learning. Six participants performed task 1 with condition A and task 2 with condition B. The other six performed task 1 with condition B and task 2 with condition A. The order of the two tasks was counter-balanced. Each participant could provide up to ten example-based pairwise constraints (both positive examples and negative examples) for each task. The active polling with uncertainty feature was disabled in both conditions. We collected each participant's task completion time for each condition and the distance metrics derived by the learning algorithm.

Active polling with uncertainty. We used a between-subjects design in this session with two conditions: the constraints & active polling condition and the constraints-only condition. Due to the nature of task, it is hard to eliminate the carry-over effect by preparing two comparable but different sets of materials when using a within-subjects design. Besides, we wanted to control the duration of the study to be one hour. The active learning feature was enabled in both conditions. The task required users to find five students with similar performances to one student named "Indrek". We told the participants that the accuracy and clarity features of the first two assignments were very important to consider and asked them to define the importance of other features themselves. We hypothesized that given meaningful clustering results, one can find similar students easily just by going over each student in the target's group. Otherwise, if the clustering results were not good, the participants would have to view groups besides the target's group to find similar students.

Free exploration. In this session, the participants were asked to group the students into three categories based on their own grouping criteria. Users were encouraged to think aloud and even write down their rules on a piece of paper. They were also encouraged to explore MindMiner as long as they wanted.

Qualitative feedback. After participants completed all the tasks, they were asked to complete a questionnaire and describe their general feeling towards our system.

3.5.2 Participants and Apparatus.

We recruited 12 participants (5 female) between 22 and 51 years of age (mean = 27) from a local university. Two were instructors from physics department and psychology department respectively. The other ten were graduate students who have teaching experience. Each study lasted for around 60 minutes (up to 90 minutes maximum), and each participant was given a $10 gift card for the time.

A Lenovo ThinkPad T530 laptop computer with Intel Core i5-3210 CPU, 4GB RAM, running Windows 7 was used. An external NEC 23 inch LCD monitor with a resolution of 1920*1080 was attached to the laptop to run MindMiner.

3.6   EVALUATION RESULTS

3.6.1   Clustering and Active Learning.

The average task completion time in the "with active learning" condition is significantly shorter than that of the "without active learning" condition (266.4s vs. 357.4s, $F_{1, 11}=13.403$, $p<0.01$). We observed that with active learning suggestions enabled, participants tended to compare the students involved, instead of randomly picking several students to compare.
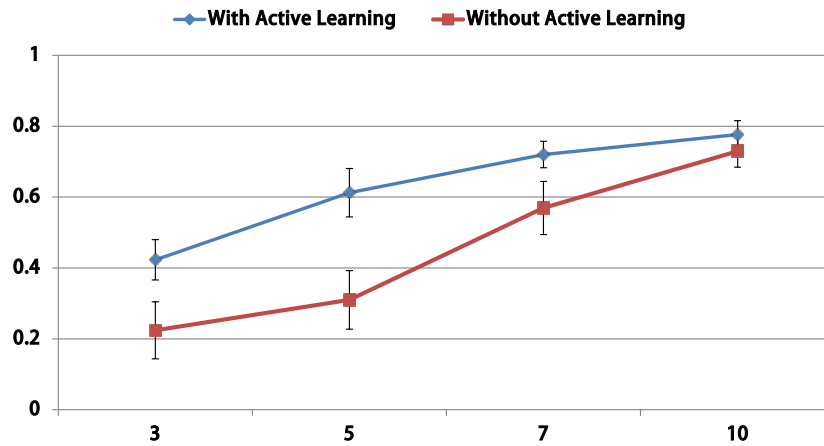
Figure 5. Average cosine similarities between "gold standard" and distance metrics learned by different numbers of constraints (the higher the better).

To evaluate the quality of distance metrics learned in the two conditions, we defined our "gold standard" to be a weight vector where the weights of predefined important features are 1s, and the weights of other features are 0s. We used cosine similarity between the standard weight vector and the weight vector learned from our algorithm to measure the quality of distance metric learned (Figure 5). Analysis of variance revealed that there was a significant difference ($F_{1, 11}=7.42$, $p<0.05$) in the quality of the distance metric learned. We found that there was a significant main effect ($F_{3, 9}=19.30$, $p<0.05$) in quality among different numbers of constraints collected.

30

Pairwise mean comparison showed that more constraints led to significantly better quality distance metrics. With the same number of constraints, the quality of distance metrics learned with active learning was significantly higher than that without active learning for all four numbers of constraints in Figure 5.

3.6.2   Active Polling with Uncertainty.

When active poling with uncertainty was enabled, the average completion time was 252.7 seconds (σ =19.6). When disabled, the average completion time was 304.8 seconds (σ=43.1). However, the difference was not statistically significant (p=0.297).

The active polling with uncertainty condition also led to significantly more similar students discovered (4.67 vs. 2.50, p<0.001) than the condition without active polling (Figure 6). This finding showed that active polling with uncertainty could also facilitate users by helping them to learning process to derive more relevant entities.
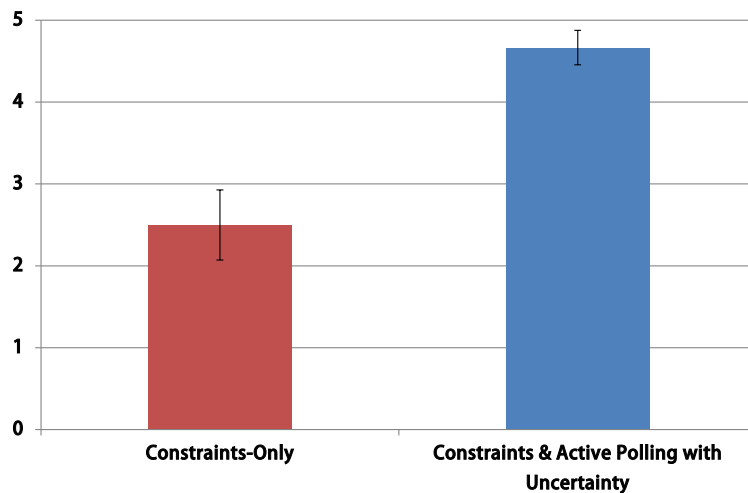
Figure 6. Average number of similar students discovered by condition (the more the better).

3.6.3 Free Exploration.

A total of 458 interaction activities were recorded in the free exploration session (Figure 7). We observed that participants tended to add more positive examples (must-link, must-belong, and similar-groups) than negative examples (cannot-link, cannot-belong, and dissimilar-groups) (78.6% vs. 21.4%) when the active learning feature was disabled. Participants tend to not provide negative examples even when they were confident that two entities were very different; when the active learning feature was enabled, the ratio of negative examples almost doubled (40.8%) and the difference was statistically significant. This observation indicated that the current active learning interface and heuristics in MindMiner can increase users' awareness and contribution to negative examples.
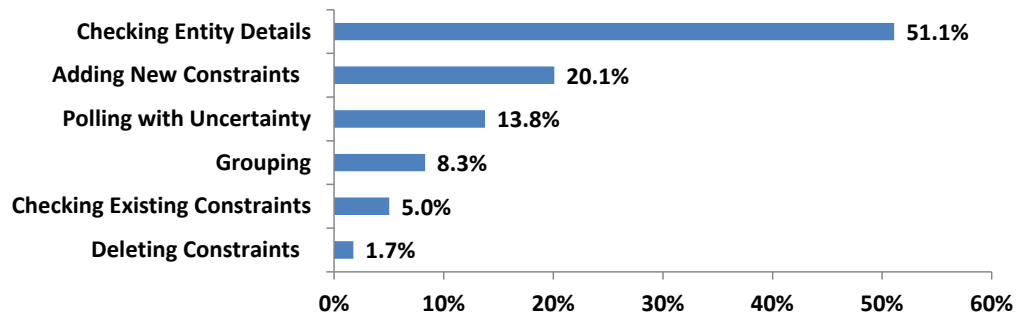
Figure 7. Activity distribution of participants.

3.6.4 Subjective Feedback.

Overall, participants reported positive experiences with MindMiner. Participants felt that the system improved their understanding of students' performance through peer-review data (Figure 8).
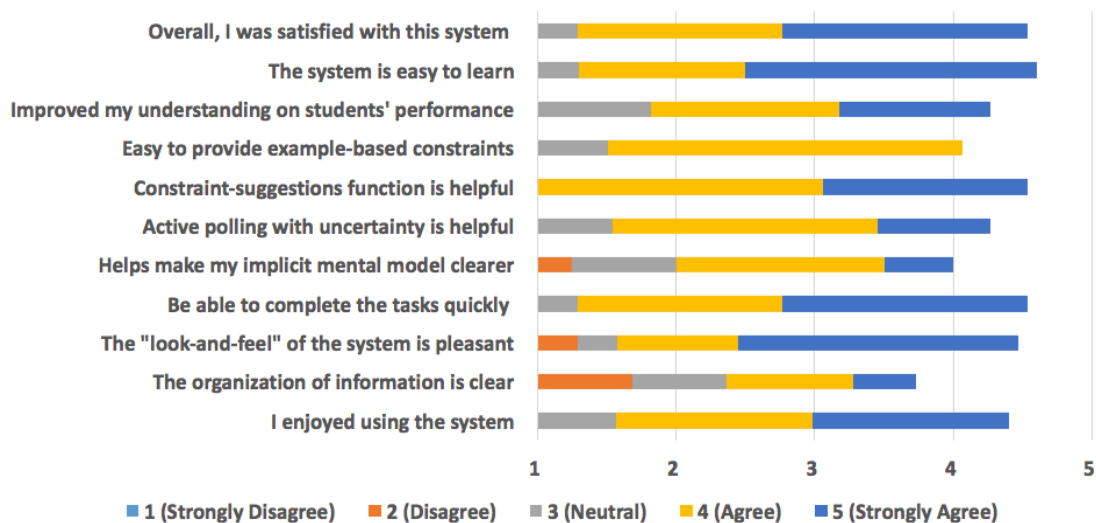
Figure 8. Subjective ratings on a 5-point Likert scale.

3.7 LIMITATIONS

MindMiner was originally designed for peer-review understanding and grading scenarios for instructors, and the size of the experiment dataset (up to 50 features, up to 150 students) is quite representative. We have not conducted large-scale deployments (e.g., involving thousands of students in MOOCs) but we believe many ideas explored in this work may be inspirational for larger-scale problems (the active learning heuristics, the active polling with uncertainty, and the result regularization step can reduce the parameter search spaces in the convex optimization stage and the number of constraints needed). Although the final clustering algorithm needs to handle all the data, the distance metric learning process can only deal with a subset of representative samples.

The current controlled lab-study and the quantitative analysis answer some fundamental questions such as whether MindMiner is easy to learn and use, etc. In the future, we will conduct longitudinal deployments to understand how well MindMiner works in the wild, with domain

experts. We expect such findings will most likely be qualitative due to challenges in enforcing controlled conditions in the wild.

3.8   SUMMARY

We presented MindMiner, a mixed-initiative interface combining visualization, machine learning and rich user interaction, to help instructors externalize their subjective domain knowledge, interactively make sense of peer review data, and improve data exploration efficiency via distance metric learning. MindMiner makes contributions in both interaction design and machine learning algorithms. In a 12-subject user study, we found that 1) MindMiner can capture the implicit similarity measurement from users via *examples collection* and *uncertainty polling*; 2) *active learning* could significantly improve the quality of distance metric learning when the same numbers of constraints were collected; 3) the *active polling with uncertainty* method could improve the task completion speed and result quality; 4) MindMiner can improved users' understanding of students' performance through peer-review data exploration.

4.0   BAYESHEART: A PROBABILISTIC APPROACH FOR IMPLICIT HEART RATE MONITORING ON CAMERA PHONES

"Time and again we have seen the disruptive impact the internet can have on industries –

driving innovation and enhancing the customer experience.

I have no doubt MOOCs will do the same for education."

— Martin Bean

4.1   BACKGROUND AND INTRODUCTION

Computer and Internet technologies are revolutionizing education. One good example is the rapid growth of Massive Open Online Courses (MOOCs), which transcends the traditional barriers of institutional access and makes the high quality learning materials available at low cost. There are around 16 million MOOC learners by the end of 2014 [150]. While it is promising in high quality education at scale, there are several challenges in MOOC learning. Firstly, students are more likely to get distractions in non-classroom environment [142]. Second, teachers can no longer monitor the students to see whether students paid attention to the lecture, and understand the material. In comparison, such information can be inferred by facial expressions, asking questions and interruptions in traditional classrooms.

To gather feedback from students about whether the online lecture was understood, researchers and instructors have explored approaches such as explicit polling or questionnaires, post-lecture reflections [66], and browser log analysis [35, 72]. However, such post-hoc analysis techniques are usually coarse-grained, with high latency, and only indirect measurements of the actual learning process. At the same time, researchers have also explored the use of various physiological signals, such as heart rates [169], galvanic skin responses [24], facial expressions [24], and Electroencephalography (EEG) [154] to infer learners' cognitive and affective states in learning. However, all of these approaches require dedicated sensors for signal collection. The cost, availability, and portability of such devices could prevent the wide adoption of such technologies in the wild. Beside that, using these devices makes the signal collection process noticeable and awkward, which is an extra and unnatural step in learning. *Then can we leverage the devices that learners already have such as camera phones, PCs, or webcams to capture and collect their physiological signals implicitly while they watch online videos?*

In this chapter we present BayesHeart [55], a probabilistic algorithm that *implicitly* extracts both heart rates and distinct phases of the cardiac cycle directly from noisy, intermittent ROI signals (e.g. fingertip transparency changes, facial color images) captured by camera phones. We use the term "*implicitly*" to differentiate our envisioned scenarios with those that require users to mount sensing equipment, "*explicitly*" launch monitoring apps, and spend an uninterrupted amount of time in data collection. For example, Figure 9 shows one of our envisioned usage scenarios in education. AttentiveLearner [131, 171] integrates lens covering gestures as video play control channel—covering the lens means play the video while uncover means pause the video play. BayesHeart can extract heart rate through the user covering actions while they watch lecture videos. AttentiveLearner further infers learner attention based on the heart rate and such

information can benefit both instructors as well as learners. In this scenario, BayesHeart can infer users' heart rates as a *side effect* during everyday mobile interactions with the back camera. Beside AttentiveLearner which can detect learners' mind wandering [131], interests and confusion [171] via implicit heart rate sensing, my colleagues at the University of Pittsburgh also create intervention technologies that can adapt to learners' perceived difficulty levels [130], as well as learners' boredom and disengagement [170] for MOOC learning. The *intelligence* involves modeling students' affective and cognitive states via implicit heart rate monitoring and enabling an efficient and scalable feedback channel from students to instructors as well as enabling personalized learning opportunities.

BayesHeart uses an adaptive hidden Markov model, requiring no user-specific training. BayesHeart has four major advantages when compared with approaches in the existing literature: 1) lower latency and bootstrap time; 2) higher accuracy under noisy and incomplete data; 3) easier integration with application scenarios that only capture ROI implicitly or intermittently [73, 172]; 4) joint extraction of both heart rate and distinct phases of the cardiac cycles.

The content of this chapter can be found in the published paper [55].
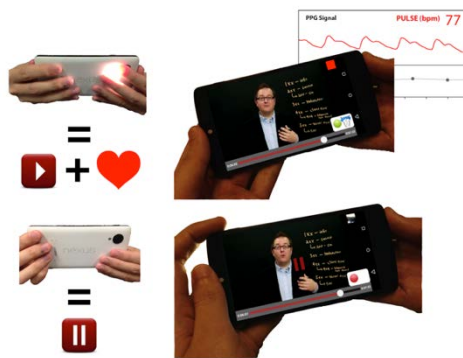
Figure 9. When integrated with MOOC mobile clients (e.g., AttentiveLearner [131, 171]), BayesHeart can detect heart rate *implicitly* from *intermittent* mobile interactions when learners watch lecture videos.

4.2   RELATED WORK

As one crucial physiological signal, resting heart rate (RHR) is a key indicator of health condition [40, 62], fitness [41, 81], and expected life span [82, 148]. Heart rate and variations of heart rate have also been used to predict human emotion [132], cognitive workload [146], stress [33, 112], and attention.

Several commodity camera based heart rate detection techniques [10, 11, 27, 65, 73, 79, 85, 128, 135, 147] have arisen in recent years. Along this line, researchers have shown the feasibility of extracting heart rate from finger transparency changes, i.e. photoplethysmography (PPG), captured by a smartphone camera [11, 85, 25]. Jonathan et al. [85] proposed to analyze fingertip video via Fast Fourier transform (FFT). Poh et al. [135] successfully inferred heart rates by analyzing facial color changes captured by a webcam. Poh's algorithm first used Independent Component Analysis (ICA) to construct a less noisy signal channel from three R/B/G channels, then used FFT and thresholding in the frequency domain for pulse counting. Similarly, Balakrishnan [10] used PCA for noise reduction, frequency domain power analysis for channel selection, and a moving window in temporal domain for peak detection when analyzing involuntary head motions in video. It's also possible to measure heart rate by analyzing facial thermal changes [65]. With the popularization of camera phones, such camera based approaches have already become wildly popular when compared with solutions that rely on dedicated hardware. For example, Instant Heart Rate [79], a commercial camera based PPG app, attracted over 25 million users within two years.

The figure is a 2D design space chart. The y-axis is labeled "Noise Reduction" with rows (bottom to top): "Static RGB Sum (i.e. Y)", "PCA", "ICA". The x-axis is labeled "Pulse Counting" with columns: "Probabilistic Modeling", "Temporal Heuristics", "Frequency Analysis".

Noise Reduction	Probabilistic Modeling	Temporal Heuristics	Frequency Analysis
ICA	*Explored in this paper*	*Explored in this paper*	Poh et al., 2010
PCA	*Explored in this paper*	Balakrishnan et al., 2013	Balakrishnan et al., 2013
Static RGB Sum (i.e. Y)	BayesHeart	Han et al., 2014	Jonathan et al., 2010

Figure 10. The design space of commodity camera based heart rate detection techniques.

Despite variations in underlying sensing mechanisms, most of today's algorithms adopt a two-step workflow, i.e. 1) noise reduction and 2) heart beat counting. The noise reduction step intends to diminish noise from digitizers, ambient light, body tissue, and motion. Commonly used noise reduction techniques include independent component analysis (ICA) [135], principle component analysis (PCA) [10], smoothing filters [11, 152], and heuristics [73]. The heart beat counting step leverages either temporal domain techniques (peak thresholding [10, 11, 152], heuristic based peak counting [73]) or frequency domain techniques (e.g., Fast Fourier Transform [10, 135]). Figure 10 shows the design space of commodity camera based cardiac pulse detection and the relationship of BayesHeart with existing techniques.

Unfortunately, although the two-step workflow works well on continual and relatively clean signals, it may break when dealing with *implicit* and *intermittent* mobile interaction scenarios (e.g., in our envisioned MOOC learning scenario, Figure 9). Figure 11 illustrates representative signals (i.e. (a) high quality signals, (b) noisy signals, and (c) intermittent signals) captured from such scenarios.
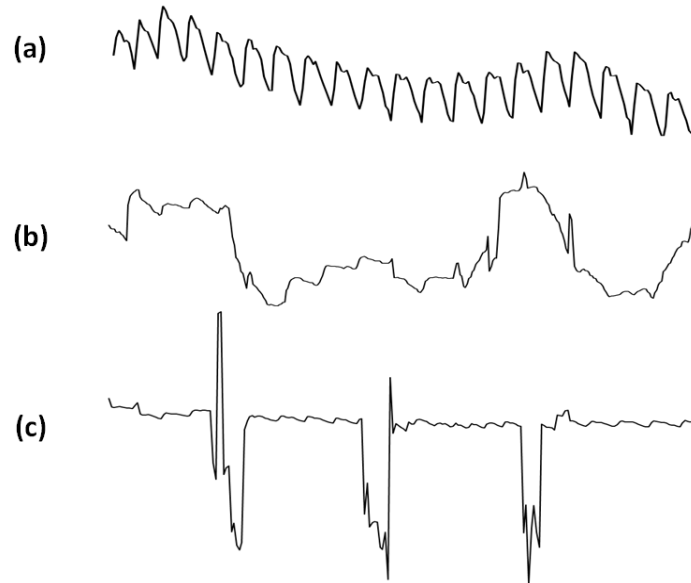
Figure 11. Sample PPG signals captured from a mobile camera (a: high quality signals; b: noisy signals; c: intermittent signals).

In the noise reduction step, component analysis techniques (ICA and PCA) require constructing and updating a linear transformation matrix from historical data. A 30-second window [135] will cause at least the same amount of bootstrap time and increased latency. The calculation of PCA or ICA transformation matrix becomes more challenging when dealing with intermittent signals that only last 5 - 20 seconds in each session.

In the heart beat counting step, an FFT based approach [135] requires *continual* signals meaning that it will break when handling intermittent signals. Meanwhile, temporal domain counting techniques only leverage the *amplitude* properties of pulse peaks and ignore the *temporal regularity* of pulse wave forms. As a result, both peak thresholding and peak counting techniques are sensitive to motion-induced noises, which are hard to eliminate during the noise reduction step.

BayesHeart uses probabilistic modeling to address challenges of existing algorithms in contexts of *noisy*, *implicit* and *intermittent* PPG signals captured by commodity cameras in everyday settings. Unique contributions of BayesHeart include: 1) The usage of an Adaptive

Hidden Markov Model to extract both heart rates and distinct phases of the cardiac cycle directly from raw signals; 2) The usage of *discrete local trend* features to achieve both simplified model training and improved robustness; 3) Designing an effective 2/4-state model selection paradigm to exploit both the *temporal regularity* and the *intra-person diversity* in signals. We also advance the state-of-the-art by identifying the design space of commodity-camera based heart rate detection and presenting a comparative study of BayesHeart, existing algorithms, and their variants.

4.3   THE BAYESHEART ALGORITHM

4.3.1   Background

The underlying theory behind photoplethysmographic (PPG) imaging is as follows: the heart pumps fresh blood to the capillary vessels of a human body during systole in each cardiac cycle. Such blood volume changes lead to changes in fingertip transparency, which can be detected by the built-in camera of the mobile phone when the user covers the lens of the camera with her fingertip [70, 71, 85]. Therefore, the changes of finger transparency can be viewed as a *generative process*, in which there are natural correlations between different regions of the PPG waveform and dedicated cardiac phase.

4.3.2   Pulse Modeling

BayesHeart relies on a hidden Markov model to capture the temporal regularity of the different stages in cardiac cycles (hidden) and the finger transparency changes (observable). After training

41

the model, given new observations, we can segment the observations into states by calculating an optimal alignment via Viterbi decoding [138]. Then heart rate can be estimated by extracting the duration of each cardiac cycle from the derived cardiac alignment.
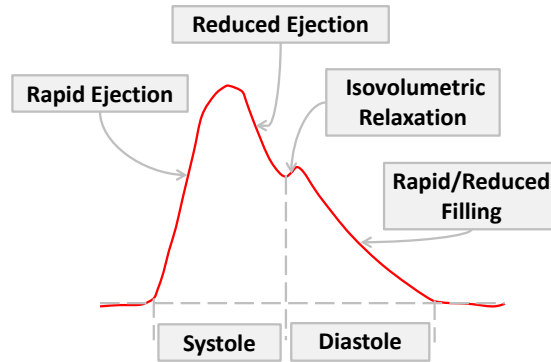
Figure 12. One-cycle waveform associated with the physical activities in one cardiac cycle.

4.3.2.1 Hidden States

According to PPG imaging [115, 152], a typical cardiac cycle includes four distinct stages (Figure 12): 1) rapid ejection (i.e. systolic upstroke); 2) reduced ejection (i.e. systolic downstroke), 3) isovolumetric relaxation (i.e. a small upstroke caused by dicrotic notch); and 4) rapid/reduced filling (i.e. diastolic downstroke). Therefore, it is a natural choice to use a 4-state hidden Markov model with each state corresponding to one cardiac stage (Figure 13.a). However, due to extrinsic noise and variations in tissue/skin reflections, the third cardiac stage (isovalumetric relaxation) can be hard to identify in the waveforms captured by commodity cameras. In such situations, the waveform of one cardiac cycle only shows two distinct phases (Figure 13.b): rapid ejection (i.e., systolic upstroke) and reduced ejection plus the whole diastole phase (i.e., a long downstroke). Hence we propose an adaptive 2/4 state model to capture both the subtlety and diversity of waveforms.

42
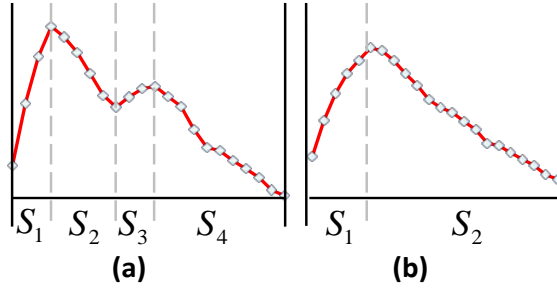
Figure 13. States selection based on the waveform shapes.

We do not consider models with more than 4 states for two reasons. 1) Although the model likelihood on training data improves as the number of states increases, it usually comes at the cost of data overfitting (i.e., more parameters) and more training samples. 2) Our main purpose is to estimate the *duration* of cardiac cycles rather than to analyze subtle changes within one cycle. For this purpose, fine grained segmentations, at the cost of more training data and increased algorithm complexity, won't bring us additional insight.

4.3.2.2 Observations

Unlike existing research on clinical ECG/PPG analysis [34, 78], we choose not to use the absolute observations (i.e. brightness of finger transparency) in our model because such absolute scales are sensitive to both the environmental illumination changes and motion-induced noise. Instead, we choose the "*local trend*" of each sample point as a more *robust* feature as our model observations. Interestingly, this feature is more *expressive* than the absolute scale in our context. For instance, the wave-form generates much more *increasing* observations in the rapid ejection stage than the filling stage. Such regularities encoded in the "*local trend*" feature are easier to capture by BayesHeart. We further define four types of discrete observations (o1 – o4) from the "*local trend*" feature (Figure 14).
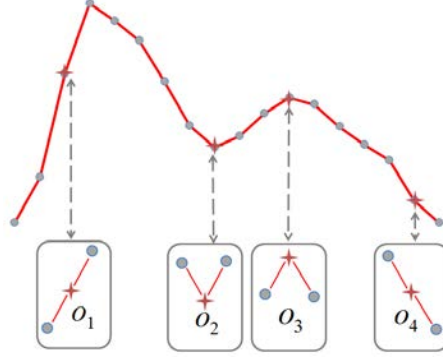
Figure 14. Four types of observations.

4.3.2.3 Mathematical formulation

BayesHeart is a discrete left-right HMM [138] defined as follows:
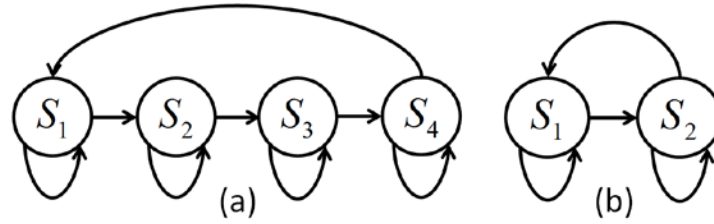
Figure 15. 4-state model (a) and 2-state model (b).

1) N, the number of states in the model

We denote the individual states as $S = S_1, \ldots, S_N$ and the state at time t as s_t. As mentioned above, there are two models (i.e., 4-state model and 2-state model, Figure 15) in our approach.

2) M, the number of distinct observation symbols per state,

In our case, M=4. We denote the individual symbols as $O = \{O_1, O_2, O_3, O_4\}$.

3) The initial state distribution $\pi = \{\pi_i\}$ where

$$\pi_i = P\{s_1 = S_i\}, 1 \le i \le N$$

4) The state transition probability distribution $A = \{a_{ij}\}$ where

$$a_{ij} = P\{s_{t+1} = S_j | s_t = S_i\}, \qquad 1 \le i, j \le N$$

In BayesHeart, we add order constraints within each cardiac cycle by setting $a_{ij} = 0$ for the (i,j) pairs in which $i > j$. The only exception is $a_{N1} > 0$ which enables the model to start new cycles.

5) The observation symbol probability distribution in state j, $B = \{b_j(k)\}$, where

$$b_j(k) = P\{o_k \ at \ t \mid s_t = S_k\}, 1 \le j \le N,$$

$$1 \le k \le M$$

The hidden Markov model in BayesHeart can be characterized in terms of three probability measures, A, B and π. For convenience, we denote the 4-state model as $\lambda_1 = (A_1, B_1, \pi_1)$ and the 2-state model as $\lambda_2 = (A_2, B_2, \pi_2)$ in a compact way.

4.3.2.4 Parameter estimation

To use the Baum-Welch method to estimate model parameters $\lambda = (A, B, \pi)$, we first estimate initial distribution of π as the temporal duration of each cardiac state in the training data. For the state transition probability distribution A, we assign a high probability (i.e., 0.8) of remaining in the same state and low probabilities (i.e., 0.2) to transitioning between states. We set $a_{ij} = 0$ (except for a_{N1}) when $i > j$ because the cardiac stages appear sequentially and cannot be reversed.

We train the 4-state model λ_1 and the 2-state model λ_2 separately and use a model selection process detailed in the next section at runtime.

4.3.3 Heart Rate Estimation

After deriving the underlying model via offline training, the BayesHeart runtime includes four phases: 1) model selection; 2) state sequence generation; 3) cardiac pulse interval calculation; and 4) post-processing.

4.3.3.1 Model selection

BayesHeart uses the first 5 seconds[1] of observations for model selection. We leverage the Bayesian information criterion (BIC) [168] to find the better model λ^* from $\{\lambda_1, \lambda_2\}$ at the same time to prevent the 4-state model from overfitting the observations.

$$BIC_\lambda = -2 \cdot \ln \Pr(o^*|\lambda) + k \cdot (\ln(n) - \ln(2\pi))$$

$$\lambda^* = \underset{\lambda \in \{\lambda_1, \lambda_2\}}{\operatorname{argmin}} BIC_\lambda$$

The model selection step does not introduce extra latency because it can be run in parallel with the state sequences generation step discussed later.

4.3.3.2 State sequences generation

In this step, we leverage the Viterbi algorithm to infer the optimal state sequence that is most likely to generate the observations by maximizing $Pr(o, s|\lambda)$.

4.3.3.3 Cardiac cycle/distinct phases extraction

We define the transition from the last state to the first state (i.e. $S_N \to S_1$) as the start of a new cardiac cycle and mark all of such transitions in the derived state sequence. Therefore, the duration d between two adjacent marks is the duration of one cycle (Figure 16). The instant heart rate estimate is:

$$Instant\ Heart\ Rate\ (bpm) = \frac{60000(ms)}{d\ (ms)}$$

Here bpm means beats per minutes. It is worth noticing that BayesHeart extracts distinct phases in each cardiac cycle in parallel with the heart rate estimation processing (Figure 16).

---

[1] We assume that owner change is rare for mobile devices and context (location, environment, etc.) change happens at the scale of minutes or hours rather than seconds. The model selection process can run more frequently when necessary.
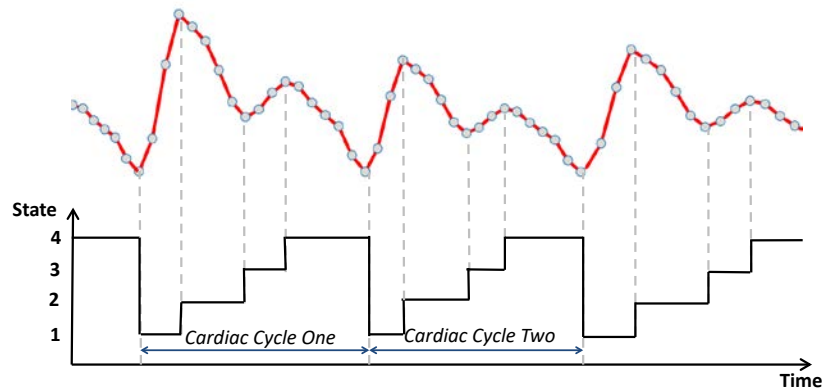
Figure 16. Cardiac cycle/ distinct phase extraction from the underlying state sequence.

4.3.3.4 Post-processing

Despite the robustness of our BayesHeart algorithm, we still impose two simple heuristics to reduce outliers in extreme situations. If either of the heuristics is violated, BayesHeart rejects the current estimation and outputs the first valid estimation in history.

Heuristic 1: *Valid heart rates are within the range of [30, 300] bpm.*

Heuristic 2: *The maximal change between two adjacent bpm estimates should not be more than (k=5) bpm.*

4.3.4   Intermittent signals

As highlighted in the introduction section, the intermittent appearance of ROI (e.g. 2 – 30 seconds) is the norm in many interaction scenarios. Therefore, we investigate how to extract heart rate via intermittent covering actions. There are three problems when dealing with intermittent covering: 1) how to detect users' covering actions (i.e., when users are covering the lens); 2) how to deal with the noise introduced by intermittent covering actions; 3) once users' covering actions can be detected, how to estimate heart rate based on several separate pieces of signals.

47

For the first problem, we leverage a fast and reliable linear classification model proposed in [172] to detect the lens covering gesture. The model uses the global mean and standard deviation of all the pixels in an image frame to infer whether the user is covering the lens or not with high accuracy (i.e., 97.9%). After this step, we get a set of data sequences (i.e., observation sequences), with each sequence corresponding to one covering action.

For the second problem, we find that most of the noise is generated by finger movements and pressure changes (e.g., at the beginning of each covering action). Therefore, we apply two techniques to reduce the extreme noise that appears in intermittent signals: 1) Discard the observation sequences corresponding to the covering actions which last less than 2 seconds; 2) For the data sequences that are longer than 2 seconds, discard the first 1 second of data for each covering action (Figure 17.a).
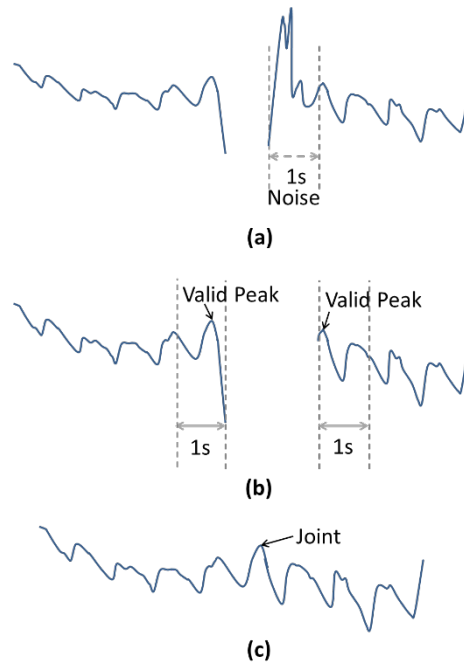
Figure 17. Additional steps with intermittent covering. a) Remove noise at the beginning of each covering action. b) Find two valid peaks. c) Connect the two valid peaks.

For the third problem, we first concatenate separate pieces of signals together to get a continuous and complete waveform. We use a heuristic to concatenate two pieces trying to make use of the data at the very end/beginning of the signal pieces. The heuristic attempts to find the last valid peak of the previous piece and the first valid peak of the next piece and link them by joining the two peaks together. Since the normal resting heart rate for adults ranges from 60-100 bpm, valid peaks are close to 600ms-1000ms apart; therefore we assume that a valid peak can be found within a window with the size of 1000ms. So we apply a window at the end of the previous piece as well as the beginning of the next piece and localize the valid peak by choosing the local-maximum point (i.e., o3) with the largest amplitude within the windows (1000ms). (Figure 17.b). Then these two valid peaks are connected together to concatenate the two signals (Figure 17.c).

4.4   EVALUATION

4.4.1   Data Collection

Although there are public datasets (e.g. PhysioBank) from medical grade ECG and PPG sensors, there is no public HR dataset for signals captured from commodity mobile cameras. Hence we collected our own dataset for this study. We collected data from 20 subjects (7 female) from department mailing lists of a local university. The participants were between 23 and 45 years old (mean $= 27.8$, $\sigma = 4.9$). We collected two types of PPG data via the built-in camera of a smartphone: 1) 10 minutes static covering and 2) 10 minutes intermittent covering. For intermittent covering, participants were asked to cover the lens for 5-10 seconds and then move their finger away for 1-3 seconds, and to repeat this process for a total of 10 minutes. Participants used one hand to operate

the mobile phone and we attached a pulse oximeter on their other hand to collect the ground truth heart rate data. Each participant was paid $5 for their time. We did not control the posture in data collection and instructed participants to "chose the most comfortable posture" at the time. 19 participants chose the sitting posture and one chose the standing posture.

We used a Google Nexus smartphone running Android 4.1 for data collection. It has a 5 mega-pixel back camera and an LED flash light. We set the built-in camera in preview mode, capturing color images of 144x176 pixel at 30 fps (frames per second). We sample 800 pixels evenly distributed in each frame and use the RGB/YUV sum of these 800 pixels to estimate the brightness of the frame. In this way we derive a set of time-stamped ROI signal vectors. We resample the data by linear interpolation to 30Hz to compensate for the jitter effect of the video stream.

The pulse oximeter in the experiment was a CMS 50D with USB port. CMS 50D is an FDA-approved, medical grade device. The accuracy of CMS 50D for pulse ratio was +/- 2 bpm.

4.4.2   Design Space Exploration

We conducted a comparative study of twelve state-of-the-art algorithms (Figure 10) in the design space of extracting cardiac pulses from commodity cameras. Such a study is important because existing literature [10, 11, 73, 85, 135] focused primarily on the *feasibility* and *non-comparative evaluation* of proposed scenarios. With the popularization of wearable devices and affect/emotion-aware intelligent mobile apps, it is imperative for researchers to gain a deeper understanding of the state of the art. To the best of our knowledge, this is the first systematic study and exploration in commodity camera based heart rate monitoring.

We explore a space of 4 noise reduction techniques by 3 pulse counting techniques for a total of 12 algorithms.

Noise reduction methods we investigated include: **1**) Using the Red channel only (baseline condition [152]); **2**) Using the Y (brightness) channel (static-weighted sum of the R, G, B channels); **3**) Using the ICA technique (dynamic-weighted sum of the R, G, B channels by maximizing channel independence); **4**) Using the PCA technique[2] (dynamic-weighted sum of the R, G, B channels by maximizing channel variance).

Pulse counting techniques include: **A**) LivePulse (temporal domain, heuristic based counting)[3] [73]; **B**) FFT (frequency domain counting, window size = 6 sec[4]); **C**) BayesHeart (temporal domain, probabilistic model based alignment).

Ignoring subtle variations in signal preprocessing and post-processing, existing algorithms can be represented as combining one noise reduction technique and one pulse counting technique. For example, the facial color based method by Poh et al [135, 134] can be represented as **3B** (their baseline condition was **1B**)**. The** facial motion based method by Balakrishnan et al [10] is **4A**[5]**.** LivePulse Games [73] used **2A** and the default BayesHeart algorithm is **2C.** Such a comparison will answer questions such as: Will the PCA/ICA based noise reduction technique be effective for detecting pulse amidst motion induced noise? Will adding a PCA/ICA based noise reduction technique improve the performance of BayesHeart even more? Quantitatively, what's the impact of an algorithm chosen in each step on the overall performance?

---

[2] We chose the most periodic channel by a method discussed in [10]. The periodicity of a signal is defined as the percentage of total spectral power accounted for by the frequency with maximal power.

[3] The LivePulse algorithm [73] is a heuristic based outlier removal and local peaks/valleys counting algorithm. LivePulse can be treated as a manually optimized, temporal domain adaptive thresholding algorithm.

[4] The choice of window size involves tradeoffs between frequency resolution, time accuracy, and latency.

[5] The original signal in [10] was head motion, but the same algorithm can be used to process PPG signals and vice-versa.

We tested the 12 combinations of algorithms with both normal and intermittent covering signals and analyzed both the accuracy and latency for each method. Considering the importance of post-processing heuristics shown in previous sections, in order to minimize confounding factors in the study, we applied the same post-processing heuristics used in the default BayesHeart algorithm in all 12 algorithms.
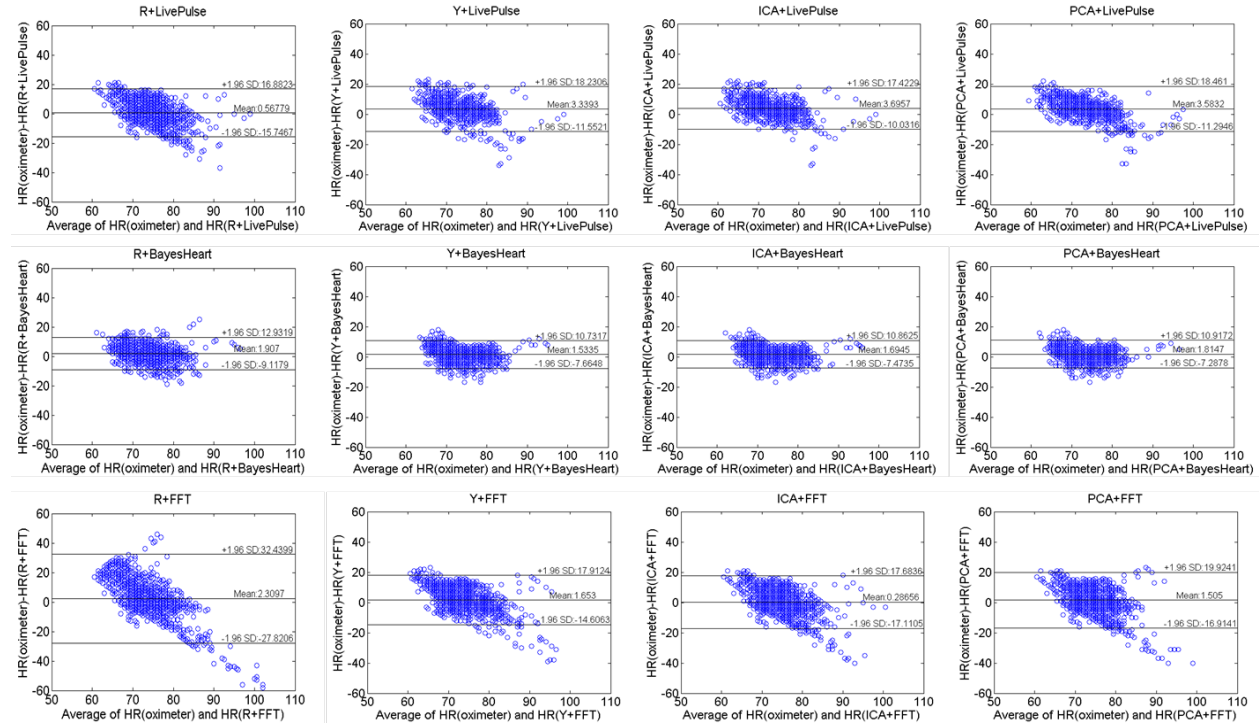
Figure 18. Bland-Altman plots demonstrating the agreement between heart rate measurements obtained from the nine state-of-the-art algorithms (with intermittent covering data) and the pulse oximeter. The lines represent the mean and 95% limits of agreement.

We use leave-one-out cross-validation (LOOCV) to test user-independent performance of BayesHeart compared with previous work. We use ***mean error rate (MER)*** to measure algorithm accuracy. To derive the MER of a given algorithm/configuration, we compare the estimated heart rate with the gold standard *every second* and report the average.

Figure 18 shows the Bland-Altman plots demonstrating the agreement between heart rate estimates generated by the 12 algorithms and gold standard with intermittent covering data. The

52

lines represent the mean and 95% limits of agreement. By comparing between different columns (i.e., different noise reduction techniques) we can see that using Y(**2**)/ICA(**3**)/PCA(**4**) could reduce the error compared to using R(**1**) directly. For example, when using BayesHeart on R channel, the mean bias was 1.91 bpm with 95% limits of agreement -9.12 to 12.93 bpm. The mean bias was reduced to 1.53 bpm with 95% limits of agreement -7.66 to 10.73 bpm when using Y(**2**). At the same time, by comparing different rows (i.e., different pulse counting methods) we can find that BayesHeart(**C**) can lower the errors compared to LivePulse(**A**)/FFT(**B**). For example, when applied on Y(**2**), LivePulse(**A**)'s mean bias was 3.34 bpm with 95% limits of agreement -11.55 to 18.23 bpm and FFT(**B**)'s mean bias was 1.61 bpm with 95% limits of agreement -14.61 to 17.91 bpm. In comparison, BayesHeart(**C**) reduced the mean bias to 1.53 bpm with 95% agreement -7.66 to 10.73 bpm.
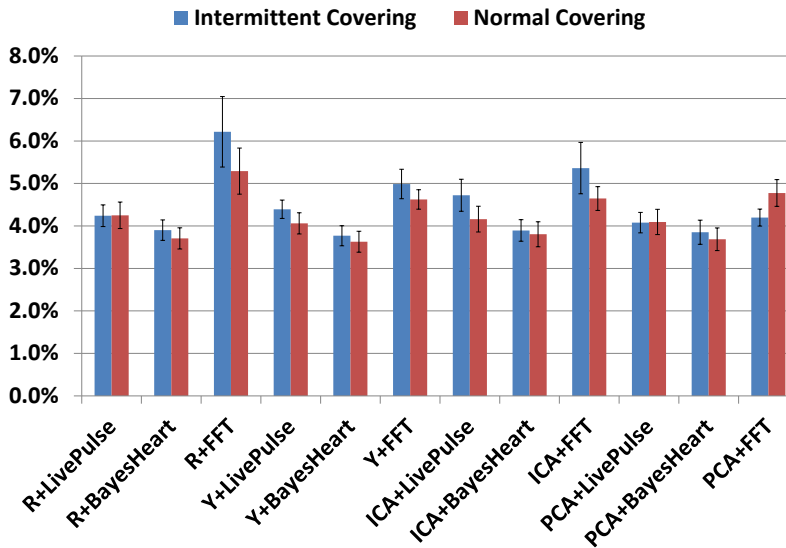
Figure 19. Mean error rates (MER) of algorithms.

Figure 19 shows the corresponding MERs. For intermittent covering, Y+BayesHeart(**2C**) has the lowest MER (3.77%), followed by PCA+BayesHeart(4**C**) (3.85%) and ICA+BayesHeart(**3C**) (3.89%). R+FFT(**1B**) has the highest MER (6.22%). For normal covering,

Y+BayesHeart(**2C**) has the lowest MER (3.63%), followed by PCA+BayesHeart(**4C**) (3.69%) and R+BayesHeart(**1C**) (3.71%). A three-way (signal type vs. noise reduction method vs. pulse counting method) ANOVA shows that noise reduction techniques ($F_{3,14}=3.36$, $p<0.05$) and pulse counting methods ($F_{2,15}=18.56$, $p<0.01$) also have a significant main effect on MERs. Signal type does not show a significant effect on MERs ($F_{1,16}=0.58$, $p=0.45$) and we attribute this to the effectiveness of three methods we proposed to handle intermittent signals.

Among noise reduction techniques, the mean error rates corresponding to Y(**2**), ICA(**3**), PCA(**4**) and R(**1**) (averaged in both signal types) are 4.24%, 4.43%, 4.11% and 4.60%, respectively. Pairwise comparisons show that both the static weighted sum approach in Y(**2**) ($p=0.04$) and the dynamic weight sum approach in PCA(**4**) ($p=0.01$) are significantly better than R(**1**) despite signal quality. Such improvements may be caused by the increased equivalent pixel area in Y(**2**)/PCA(**4**). However, the difference between the MERs of ICA(**3**) and R(**1**) is not significant ($p=0.39$). Although PCA(**4**) has a lower MER than Y(**2**), the difference is not significant ($p=0.13$). We attribute that to the non-linear nature of skin/tissue reflection and the latency involved in calculating the transformation matrix, which was in turn used to capture the dynamic nature of extrinsic noises.

Among pulse counting techniques, the mean error rates corresponding to LivePulse(**A**), FFT(**B**), and BayesHeart(**C**) (averaged in both signal types) are 4.25%, 3.78% and 5.01%, respectively. Pairwise comparisons show that BayesHeart can lower MERs significantly compared with LivePulse ($p<0.001$). And LivePulse can also significantly lower MERs compared with FFT ($p=0.002$). The reasons include: 1) the heuristic based method (e.g., LivePulse), although simple, could not capture both the diversity and regularity of signal with dealing with increased amount of signals; 2) FFT is the most sensitive technology to noise; the low-sampling rate (30HZ) could be

one reason that led to the bad performance of FFT. Besides, the fixed-size window in FFT may also introduce increased noise when dealing with brief, highly intermittent signals; 3) BayesHeart exploits additional information in trellis structure, the state transition cost, and temporal regularity in signals through a simple yet robust probabilistic model; such increased "*signal/noise ratio*" becomes critical when dealing with increased extrinsic noise.

For pulse-counting methods, pairwise comparisons show that LivePulse and BayesHeart can significantly lower latencies when compared with FFT ($p<0.01$). This is because the relatively small number of signal samples and low sampling rate have a negative impact on the frequency domain resolution of FFT. Such low frequency domain resolution leads to less accurate estimates. Therefore, the corresponding algorithms require more samples in order to derive accurate estimates.

4.4.3 Limitations

Currently BayesHeart focuses on extracting HR from mobile interactions (Figure 9) and augmenting interactions by inferring stress/attention levels from HR readings; we leave the adoption and evaluation of BayesHeart in contexts such as exergames and healthcare to future work. Besides, we restricted our ROI to fingertip transparency captured by the back-camera in this research. By feeding output from a Viola-Jones face detector in OpenCV, the same algorithms can extract HR from facial color or facial motion.

4.5   SUMMARY

In this chapter we present BayesHeart, an accurate, low-latency probabilistic approach for implicit heart rate monitoring via commodity cameras. When integrated with MOOC mobile client, BayesHeart can be used to capture and collect learners' heart rates and infer their stress level, cognitive workload, as well as attention. We demonstrated both the feasibility and the quantitative performance of BayesHeart to measure heart rate via camera phone. In a 20-subject experiment, we systematically evaluated the state-of-the-art algorithms covering the design space regarding accuracy and latency performance. We released the source code of BayesHeart under BSD license at http://mips.lrdc.pitt.edu/bayesheart.

5.0 COURSEMIRROR: SCALING REFLECTION PROMPTS IN LARGE CLASSROOMS VIA MOBILE INTERFACES AND NATURAL LANGUAGE PROCESSING

"By three methods we may learn wisdom:

First, by reflection, which is noblest;

Second, by imitation, which is easiest;

and third by experience, which is the bitterest."

—Confucius

5.1 BACKGROUND AND MOTIVATION

The degree and quality of interactions between students and instructors are critical factors for students' engagement, retention, and learning outcomes [137]. However, such interactions are limited in large classrooms (e.g., undergraduate level introductory STEM courses) and online courses. It is safe to predict that the issue of class size will only get worse due to enrollment increase (e.g., undergraduate enrollment increased by 46% from 1990 to 2013 [8]) and educational budget cuts [120].

In recent years, researchers in education have discovered the feasibility and effectiveness of *"reflection prompts"* [16] (a.k.a. *"muddy cards"* [123] or *"one-minute papers"* [75]) to

improve both teaching and learning across multiple disciplines. In a typical deployment of *reflection prompts*, students are given index cards at the end of each lecture and are encouraged to reflect on what was confusing in the lecture. After collecting responses from students, the instructor summarizes the student reflections, identifies major misunderstandings, and plans follow-up actions, such as providing feedback in the following lectures, and tailoring the teaching plan in the future. Previous studies in different domains [99, 9, 16, 117] consistently confirmed that reflective activities could benefit students by enhancing their retention and comprehension in learning.
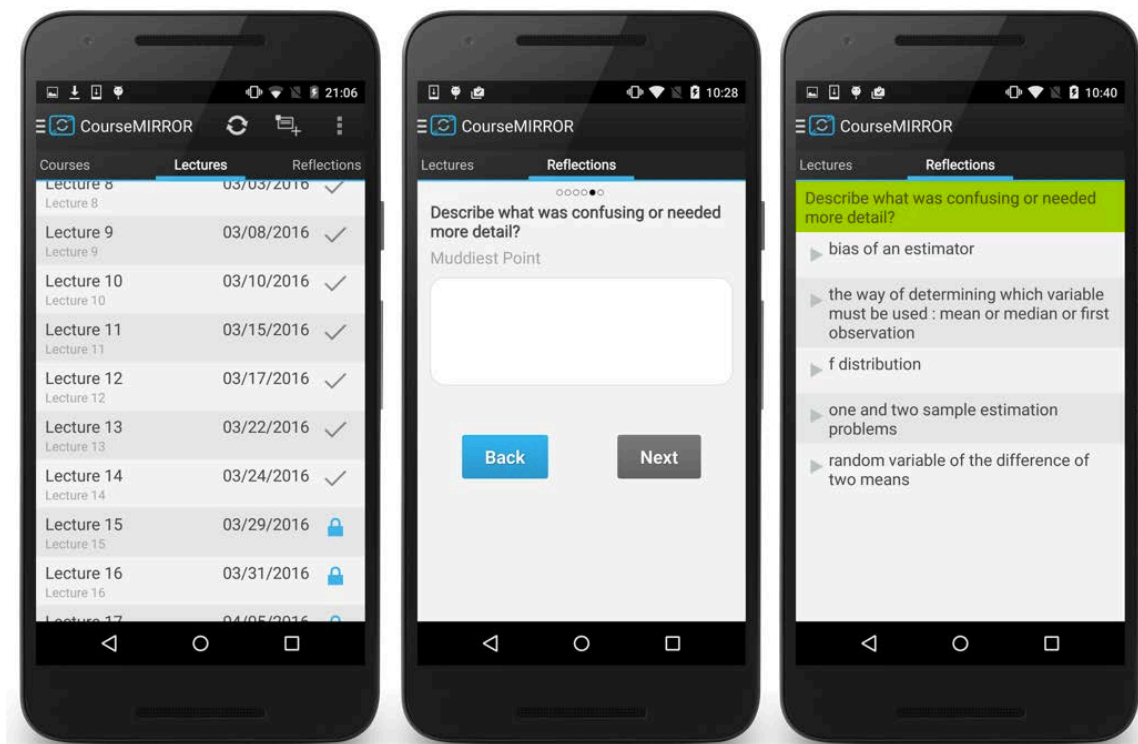
Figure 20. CourseMIRROR interfaces. a) lecture list; b) a sample reflection prompt; c) reflection summary page.

Despite the simple workflow and the encouraging efficacy, there are at least three key challenges when deploying *reflection prompts* in large classrooms. First, it is tedious and time consuming to remind and collect students' reflective responses after each lecture. Second, it is also time consuming for instructors to summarize and make sense of the raw response data [123]. Third,

as highlighted by Fan et al [58], it is difficult to maintain students' *sustained motivation* to compose concrete, specific and pedagogically valuable reflections through multiple months.

In this chapter, we present the iterative design, prototype, and evaluation of CourseMIRROR[6] [58, 59, 109] (<u>M</u>obile <u>I</u>n-situ <u>R</u>eflections and <u>R</u>eview with <u>O</u>ptimized <u>R</u>ubrics, Figure 20), a mobile learning system that uses natural language processing (NLP) techniques to enhance large classroom instructor-student interactions via streamlined and scaffolded *reflection prompts*. CourseMIRROR can 1) remind students to submit in-situ written reflections after each lecture, and collect such reflections in a scalable manner; 2) continuously monitor the quality of the reflection in composition time and generate engaging and helpful feedback to scaffold reflection writing; 3) summarize the gist of reflections and present the most significant ones to both instructors and students. The *intelligence* of CourseMIRROR involves the interface having knowledge of the domain (i.e. student reflection) and providing personalized writing scaffoldings for students and generating relevant summarizations for instructors. Through a combination of a 60-participant lab study and eight semester-long deployments involving 317 students, we found that the reflection and feedback cycle enabled by CourseMIRROR are beneficial to both instructors and students.

The content of this chapter can be found in the published papers [58], [59], and [109].

---

[6] Mobile apps for Android and iOS platforms and a mobile HTML5 optimized web version are available for free at http://www.coursemirror.com

5.2   RELATED WORK

5.2.1   Reflections in Learning

Reflection is a key component of self-regulated learning [21]. It is a fundamental learning activity in which people *"recapture their experience, think about it, mull it over and evaluate it"* [15]. Previous research illustrated the value of learners' reflection on what they had done, processed or engaged in [99, 16], as well as on their confusing (i.e. *muddy*) points [117]. Studies also suggested that reflection could benefit students by helping them identify the misconceptions in their current beliefs [31, 107] and enhance their retention and comprehension in learning [99], even without external feedback [167]. Williams and colleagues [167] found that prompting and encouraging students to *explain* abnormal corollaries (e.g. people receiving lower *absolute* grades in exam A could have higher *relative* performance than those in exam B) were more effective than asking students to describe a concept.

Traditional implementations of *reflection prompts* via *muddy cards* [123] and *one-minute-papers* [75] can face scalability problems in large classrooms. As reported by Mosteller [123], it took an instructor 30-45 minutes to summarize reflections from a 50-student class. Moreover, recklessly composing *any* reflection is insufficient for effective learning—the quality also matters. Menekse et al [117] related the characteristics (e.g., the details included and the cognitive processes identified) of students' daily reflections to Chi's iCAP framework [30] (i.e. *passive*, *active*, *constructive* and *interactive* learning activities). By analyzing and coding the reflections based on a quality rubric (Figure 21), Menekse and colleagues [117] observed a significant positive correlation between the quality of reflections (i.e. *none*, *vague*, *general* and *specific*) and the learning gains. CourseMIRROR goes beyond a mobile implementation of *reflection prompts* by

facilitating and scaffolding the composition and dissemination of reflection prompts via intelligent user interfaces.
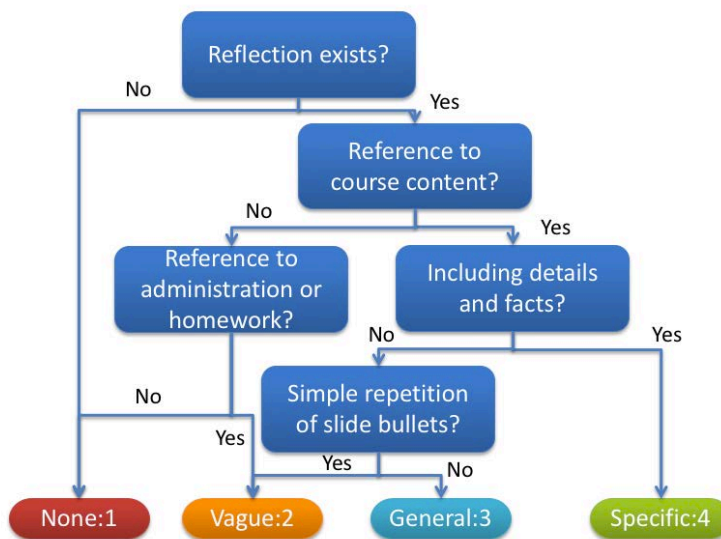
Figure 21. Rubric of reflection quality [117].

5.2.2   Computerized Reflection and Feedback Collection

Instructors in traditional classrooms can leverage audience response systems (ARSes, a.k.a. *"clickers"*) [22, 38] to collect real-time responses from students. However, ARSes are designed for multiple choice questions (MCQ) or True/False questions rather than open-ended reflections. Moreover, the hardware requirements and cost issues could prevent the widespread adoption of such systems.

Researchers also proposed various analytic techniques [22, 38, 95, 94, 171] to gain insights into student activities in MOOCs and flipped classrooms by analyzing artifacts generated in the learning process. For example, instructors can infer confusions and misconceptions of students by monitoring online discussion forums [22, 38], analyzing students' interaction logs [94], embedding

and reviewing in-video exercises [95], and detecting students' cognitive states by mining their physiological signals [171].

Mudslide by Glassman et al [66] allows students to spatially anchor their confusions as circular "*muddy points*" directly on lecture slides and visualizes the aggregated annotations to instructors. Although both Mudslide and CourseMIRROR can scale the *"muddy cards"* workflow, there are major differences between the two systems beyond target platforms (i.e. PCs vs. mobile). First, Mudslide is optimized for video watching in online courses and flipped classrooms, whereas CourseMIRROR reminds and collects students' reflections *in-situ* in traditional large classrooms. Second, Mudslide relies on lecture slides to localize confusions of students spatially. In comparison, CourseMIRROR distributes *open-ended* prompts and leverages interactive *scaffolding* to help students to compose high quality reflections in *natural language*. Third, CourseMIRROR uses text summarization algorithms to capture the gist of student responses while Mudslide leverages point cloud style visualizations to help instructors quickly locate confusions in the lecture slides.

5.2.3   Mobile Survey and Experience Sampling Methods

Through a study with 1,500 U.S. panelists, researchers found that mobile phone participants were willing to provide short responses to open-ended questions [165]. Multiple research projects also confirmed that mobile phones can be viable and comparable devices for short and optimized surveys [20, 18].

Reflection collection is also relevant to the Experience Sampling Method (ESM) [101] and Diary Studies [28] in HCI. Although systems such as Momento [28] and MyExperience [63]

support event-contingent ESM via either SMS or context-activated polling, they were not designed and optimized in educational settings.

5.3   DESIGN OF COURSEMIRROR

There are four major design goals for CourseMIRROR:

G1: Provide students a convenient and efficient way to compose and submit reflection responses *in-situ*.

G2: Encourage and help students to create specific and pedagogically valuable reflections.

G3: Facilitate instructors to make sense of students' written reflections efficiently in large classrooms.

G4: Assist students to read their classmates' reflections for peer learning
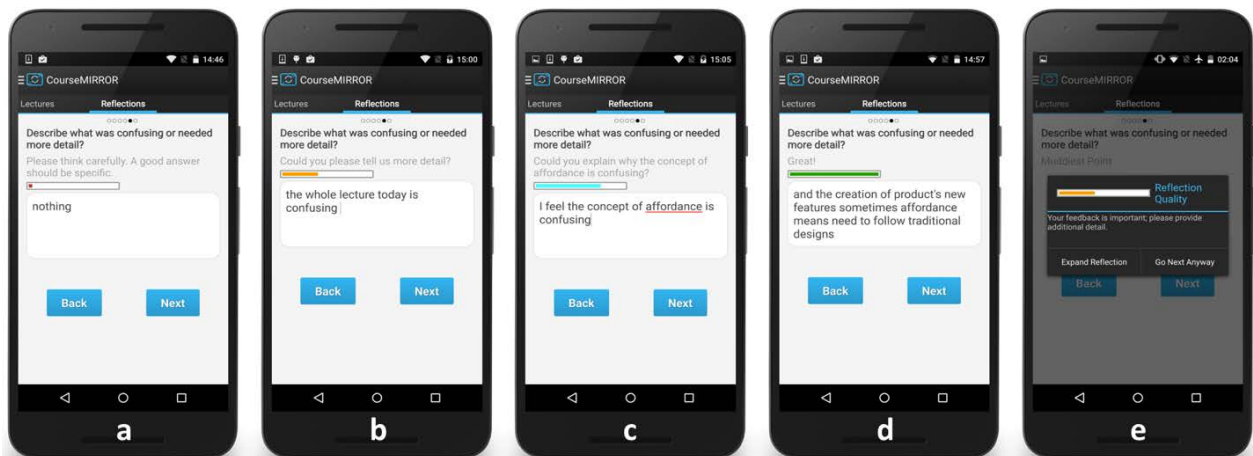
Figure 22. Reflection writing interfaces with quality feedback. a, b, c, d) instant feedback (IF, appear constantly at composition time); e) latent feedback (LF, appear as a dialog box after a submission attempt).

CourseMIRROR is designed as a mobile app to fulfill **G1**. A recent survey [44] indicates 92% of undergraduates in the U.S. own smart phones. The *instant on*, *always connected* abilities

of mobile devices could allow students to compose and submit reflections efficiently. Further, CourseMIRROR sends automatic, lecture time-triggered push notifications to collect students' reflections *in-situ*.

In order to fulfill **G2**, CourseMIRROR continuously monitors the quality of the reflection at composition time and generates engaging and helpful feedback to scaffold reflection writing (Figure 22). This design was inspired by recent research findings that providing context-sensitive feedback on students' self-explanations could help them construct better explanations when using intelligent tutoring systems (e.g., in Cognitive Tutor [3] and SE-COACH [39]).

To achieve **G3**, CourseMIRROR runs customized automatic text summarization algorithms on the server to generate a summary after each lecture (Figure 20.c). We hypothesize that relevant and coherent summaries can help instructors quickly identify students' confusion and misunderstandings. To realize **G4**, CourseMIRROR also allows students to share and access the summaries with their classmates. We hypothesize that reading reflection summaries can benefit students by letting them revisit and reevaluate the learning contents from different perspectives.

5.3.1 Text Summarization Algorithm

We explored word level, phrase level and sentence level summarization techniques and chose phrase level summarization after pilot tests. We found phrases are easy to read and browse just like keywords, and can fit better on small devices than sentences. Phrase level summarization also provides more coverage than sentence level summarization under a given length limit.

CourseMIRROR utilizes the text summarization algorithm proposed by Luo et al. [111], which was specifically designed for the purpose of summarizing reflective responses from students. This algorithm emphasizes both the *representative* (high frequency reflections) and the *diversity*

of the students (who wrote the reflections). It consists of three steps. First, use a syntax parser to generate candidate noun phrases since the knowledge concepts are usually referred as noun phrases. Second, cluster the candidate phrases into groups via the K-Medoids algorithm based on similarities of the semantic meaning. The algorithm measures semantic similarity between phrases via Latent Semantic Analysis [49]. With relevant clustering, the algorithm addresses the lexical variety problem (e.g., students use different words *"bicycle parts"* and *"bike elements"* for the same meaning). Third, select the most representative phrase in each cluster via a graph-based ranking model (i.e. LexRank). The selected phrases are then re-ranked by the number of students who mentioned the phrases. Phrases mentioned by more students should receive more attention from the instructor. This algorithm was evaluated on an engineering course corpus provided by [117], and achieved a significantly better performance in terms of ROUGE scores than a variety of other algorithms, such as MEAD, LexRank, and MMR.

5.3.2   Interactive Reflection Quality Feedback

In two pilot deployments of an early version of CourseMIRROR, Fan et al [58] found that some students began to submit brief and trivial reflections (e.g., *"none"*, *"N/A"*, *"all good"*) after months of extended use. Such reflections were neither informative to instructors nor beneficial in learning. Meanwhile, the length of reflections decreased significantly over time (12.3 words in the first half of the semester vs. 9.9 words in the second half of the semester [58]). Such findings highlight the challenges in 1) maintaining the *sustained motivation* for students throughout a semester; and 2) encouraging students to compose *high quality* reflections. Similar problems also existed in traditional intelligent tutoring systems, e.g., Aleven and colleagues [4] observed that

students provided very few explanations and even fewer good explanations when using an intelligent tutor that only prompted for explanations.

We have designed and implemented a novel quality feedback feature (Figure 22) in CourseMIRROR to address, at least in part, these two challenges. When a student is composing a reflection, CourseMIRROR continuously monitors the quality of the reflection and generates encouraging and informative feedback to scaffold the reflection writing process. The feedback is provided via a color-coded progress bar and improvement suggestions in natural language. The progress bar (Figure 22.a-22.d, above the reflection edit box) creates a visual of the quality of the current reflection in composition. A full progress bar indicates that the reflection is specific and detailed. This metaphor could inform students of how close they are to creating high-quality reflections. The improvement suggestions in natural language are also shown above the progress bar. Such suggestions give students specific, easy to follow instructions on *how* to improve the quality of their current reflection. This design is in part inspired by findings on providing feedback in intelligent tutoring systems [3, 39] and peer review systems [125]. Researchers found that *context-sensitive* feedback can help students construct better explanations to their solutions, even when the feedback is very simple (e.g., the correctness of the explanations [3]). Previous study also suggested that providing feedback regarding the presence of solutions to students could help them generate more comments with solutions in peer reviews [125].

We explored two different timings to deliver quality feedback by designing both an *instant feedback* (**IF**) feature and a *latent feedback* (**LF**) feature. Instant feedback (Figure 22.a-22.d) is always visible to students during the composition process. Latent feedback (Figure 22.e) appears in a dialog box after clicking *"next"* or *"submit"* button. Students can choose either to go back and revise the draft or to submit the reflections after receiving the latent feedback.

66

5.3.2.1 Reflection Quality Prediction

CourseMIRROR extended the classifier-based approach proposed by Luo [110] to predict reflection quality based on the rubric in Figure 21. The original quality classifier [110] uses a Support Vector Machine (SVM) with linear kernel. Features include unigram (i.e. whether a word is present), word count, and part-of-speech (e.g., whether a proper-noun is present). These features are also widely used in other NLP tasks including automatic text scoring [140] and text classification [145], and are proven to be effective. The classifier was trained on previous student reflection datasets [117] containing 1,257 reflections and the corresponding expert-rated quality scores.

Although the quality classifier above can achieve good accuracies on pre-collected reflection corpora, the classification accuracies drop significantly when classifying reflections from new courses with very different vocabulary and learning topics when compared with the training courses. The domain miss-match problem (i.e. cold start) is commonly acknowledged in various natural language processing applications, such as text classification [45], sentiment classification [127], and part-of-speech tagging [6]. In practice, it would be impossible to collect and annotate reflections for each new course, and then train a course-dependent quality classifier.

To address this challenge, CourseMIRROR uses a combination of a statistical NLP classifier and three complementary *pattern matching* techniques (Figure 23) to achieve high accuracy and more relevant reflection quality prediction in a course-independent manner. The three pattern matching techniques include *domain words matching*, *categorical patterns*, and *quality patterns*.

Domain words matching is based on an exhaustive list of domain words extracted from the lecture slides[7]. It is a reasonable assumption that reflections with domain words are at least on-topic and relevant. Thus it introduces domain knowledge for the quality prediction.

Categorical patterns are the frequently appeared exemplar patterns in each quality category. For example, *"N/A"*, *"nothing"*, and *"all good"* are categorical patterns of "none (1)" reflections while a simple repetition of a slide title is a categorical pattern of "vague (2)" reflections.

Quality patterns include *abstract* phrase and word level signals for both high and low quality reflections. They are independent from specific course topics. For instance, starting with *"what/how/why"* and ending with *"?"* typically indicates that the input is a concrete question, which is a sign of high quality reflections. In comparison, the words *"everything"* or *"the whole lecture"* usually lead to vague expressions, and thus they are signs of low quality reflections.

By analyzing the expert-annotated student reflection dataset [117], two researchers iteratively generated a total of 15 *categorical patterns* and a total of 33 *quality patterns*. Table 4 shows some sample patterns.

---

7 Although CourseMIRROR maintains a course-dependent domain word list for each course, the NLP classifier in CourseMIRROR no longer requires course-specific training.
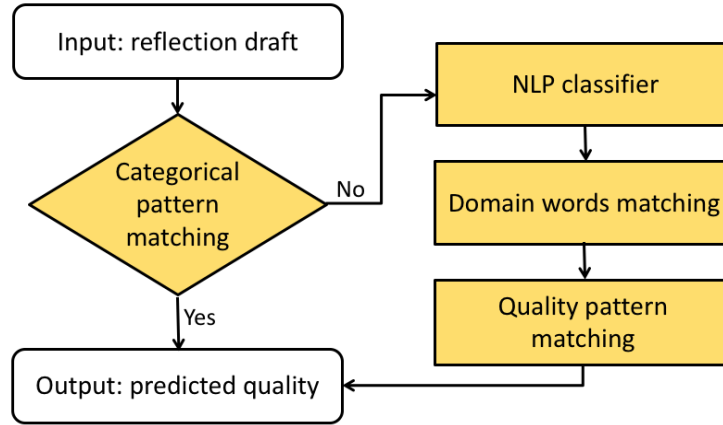
Figure 23. Workflow of reflection quality prediction.

Figure 23 illustrates the overall workflow of the reflection quality prediction algorithms in CourseMIRROR. When a reflection matches any *categorical pattern* during runtime, the algorithm directly outputs the corresponding quality score without invoking the NLP classifier. This branch reduces both the computational power and network bandwidth. Otherwise, the NLP classifier first predicts the reflection quality, then the predicted quality score is adjusted according to the results of *domain words matching* and *quality pattern matching*, according to **eq. 1** below:

$$Q = q + \alpha * DW + \sum_{p_i \in QP} p_{i_{weight}} \qquad \textbf{(eq. 1)}$$

Here q represents the classifier-predicted quality, *DW* is the number of matched *Domain Words*, α is the weight (i.e. 0.5), *QP* is the set of matched *Quality Patterns*, and p_{i_weight} is the weight (range from -1 to 1) of the particular pattern p_i.

The three complementary patterns are implemented as database tables of regular expressions on the server side. In addition to global patterns, CourseMIRROR also allows instructors to define and customize course-specific patterns and improvement suggestions.

5.3.2.2 Improvement Suggestions (Hints) Generation

CourseMIRROR provides encouraging and specific improvement suggestions based on the predicted quality (i.e. none, vague, general, specific) and the actual contents of the reflection. For example, when a student writes a "none" reflection, the system asks her to *"think carefully and start by naming a concept that is difficult to understand"*. When a student writes a "general" reflection, the system asks her to *"be more specific and tell us why you feel confused"*. CourseMIRROR pre-loads multiple hand-crafted sentences as candidate suggestions for each category, and randomly selects one from the corresponding group to maintain the feedback diversity.

Table 4. Pattern matching examples.

Category	Examples	Action
Categorical Patterns	*"nothing", "N/A", "all good"*	Output the category
Quality Patterns (positive or negative)	*"...what/how...?"* *"...relationship between..."*	Added as a new entry in Equation 1 (third component)
Domain Words	*"..affordance..", "...p value..."*	Added as a new entry in Equation 1 (second component)

By supporting the *capture group* feature in regular expressions, CourseMIRROR can detect, extract specific concepts (e.g. *affordance*) in reflections and refer to them in the improvement suggestions. For example, when CourseMIRROR detects that the input pattern is *"[X] is confusing"* (where *[X]* is a concept in the lecture), it then generates the hint *"please explain *why* [X] is confusing"*. In this way the system could generate more relevant and specific feedback based on the semantic meaning or the structure of the input.

5.4 LAB STUDY

5.4.1 Study Design

We conducted a 60-participant lab study to investigate the usability and efficacy of the interactive reflection quality feedback feature. We applied a between-subjects design with three conditions: *No-Feedback* (**NF**), *Latent-Feedback* (**LF**), and *Instant-Feedback* (**IF**). Under **NF** condition, participants write reflection without any feedback from CourseMIRROR. In comparison, CourseMIRROR provides both quality feedback and textual hints under both **LF** and **IF** conditions.

During the study, participants watched 3 short lecture videos (7-10 minutes each) from the "Model Thinking" course by Prof. Scott Page in University of Michigan [126]. After finishing each lecture, participants responded to the following reflective questions on CourseMIRROR:

- Learning Point: "What have you learned in today's class?"

- Muddy Point: "What was confusing in today's class?

At the end of the study, we conducted semi-structured interviews to solicit participants' subjective feedback on the interactive quality feedback design. We aimed to gain further understanding about how the feedback on reflection quality was perceived and digested and how it affected the writing process.

5.4.2   Participants and Apparatus

We recruited 60 participants (25 female) between 19 and 36 years of age (mean=27) from a local university, who were randomly assigned to the three conditions. The study lasted for around 60 minutes, and each participant received $10 for their time.

The participants watched the lecture videos on an Apple iMAC, with a 1.6GHz dual‑core Intel Core-i5 processor, 8 GB RAM, and a 21.5-inch display. We used a Samsung Galaxy Note 3 smartphone with a 5.7-inch display running Android 5.0 for the CourseMIRROR mobile client.
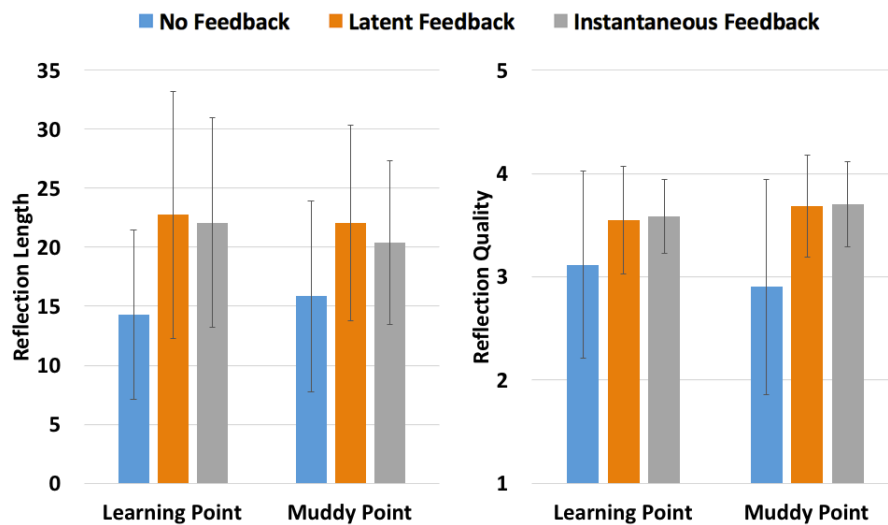
Figure 24. Reflection length and quality by reflection question. Error bars show one standard deviation.

5.4.3   Experimental Results

5.4.3.1 Quality feedback can help participants create longer and higher-quality reflections.

We chose word count as the first quantitative metric to understand reflections collected. As shown in previous research such as Experience Sampling Method (ESM) [101] and creative editing [13], word count is a good marker of writing quality because it correlates indirectly with the number of

details included. As shown in Figure 24, left, the average reflection length was 15.07, 22.40, and 21.24 words for the **NF**, **LF**, and **IF** condition respectively. Analysis of variance results showed that there was a significant difference ($F(2, 57)=5.64$, $p<0.01$) in reflection length. Pairwise mean comparison (t-tests) showed that the reflection length between **NF** and **IF** ($t(38)=2.95$, $p<0.05$, $d=0.94$)**, NF** and **LF** ($t(38)=3.01$, $p<0.01$, $d=0.95$) were significantly different. There was no significant difference in reflection length between **IF** and **LF** ($p=0.62$). Question type (learning point, muddy point) did not exhibit a significant effect on reflection length ($F(1, 57)=0.07$, $p=0.79$).

We recruited two raters to give independent quality ratings of the reflections based on the rubric in [117] (Figure 21). The agreement between the two raters was high (percent agreement: 85.0%; Cohen's kappa: 0.72; Quadratic Weighted Kappa[8]: 0.91). Disagreements were settled by discussions between the two raters after the independent coding sessions. As shown in Figure 24, right, the average reflection quality was 3.01, 3.62, and 3.64 for the **NF**, **LF**, and **IF** condition respectively. Analysis of variance results showed that there was a significant difference ($F(2, 57)=12.63$, $p<0.001$) in reflection quality. Pairwise mean comparison (t-tests) showed that the reflection quality of **IF** was significantly higher than **NF** ($t(38)=4.56$, $p<0.001$, $d=1.43$), the reflection quality of **LF** was significantly higher than **NF** ($t(38)=3.93$, $p<0.001$, $d=1.22$). There was no significant difference in reflection quality between **IF** and **LF** ($p=0.22$). Question type did not exhibit a significant effect on reflection quality ($F(1, 57)=3.34$, $p=0.073$) either.

---

[8] Since the quality scores are ordered, incorrect predictions have different costs (e.g., predicting "3" as "1" is more severe than predicting "3" as "2"). Therefore, we also report Quadratic Weighted Kappa.
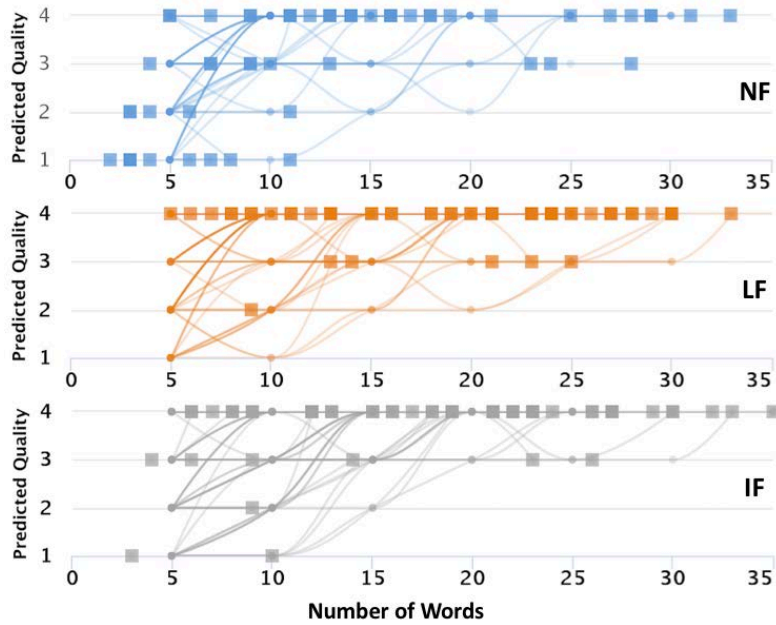
Figure 25. Predicted reflection quality by writing progress (i.e. words completed). Small dots denote the predicted quality at corresponding length. Square symbols represent submission attempts by learners.
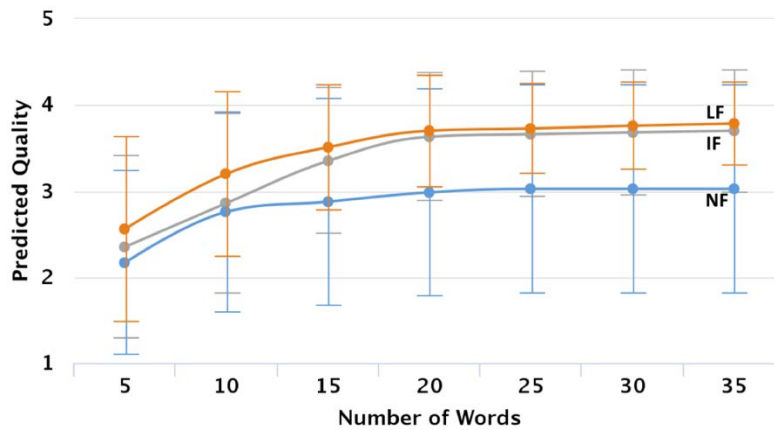
Figure 26. Average predicted quality scores by condition and writing progress (i.e. words completed).

To gain further understanding of the impact of feedback type on the reflection composition process, we plotted the predicted quality scores for each submission (Figure 25) and average performance (Figure 26) when reflection was in composition at different lengths. Please note that the quality feedback was invisible to participants in **NF** condition and was also not visible to participants in **LF** condition before a submission attempt. From both Figure 25 and Figure 26, we can clearly observe that participants in **NF** condition tended to submit reflections early, with lower

quality when compared with participants in **LF** and **IF** condition. In the **LF** condition, due to the lack of quality feedback before a submission attempt, it was more common to encounter *decreases* in predicted quality in the middle of composition when compared with the **IF** condition (Figure 25). Overall, 60% of the reflections in **NF**, 88.3% of the reflections in **LF**, and 85% of the reflections in **IF** received the highest quality rating (Specific:4, Figure 21) when participant finalized their compositions. At the same time, participants in **NF** condition submitted more vague:2 or none:1 reflections (13.3%) than those in **IF** (3.3%) and **LF** (0%) condition.

5.4.3.2 Qualitative results on instant quality feedback (IF)

Participants in **IF** reported that the progress bar made them feel *"mental pressure" [S20]* and *"obligated to fulfill the bar" [S19]* while writing reflections— *"this feature act like a supervisor that stared at me to force me to do a better work" [S1]*. At the same time, they got *"the feeling of achievement" [S4]* and were highly encouraged when they saw their progresses:

- *"It gives you a hint about how is your feedback's quality and it feels like a reward to gain full credit for feedback." [S14]*

- *"it is pretty satisfying to see the bars filling up—it is quite encouraging" [S1]*

Participants also reported that the improvement suggestions in natural language were helpful in guiding them to create deeper reflections.

- *"It tells you specifically what you should improve on" [S9]*

- *"At first I just wrote some topic words, but I saw the quality is low and it asked me to illustrate why the concept is confusing. This can definitely make me think deeper." [S1]*

To our surprise, two participants reported that sometimes the progress bar metaphor could be discouraging—they stopped thinking and writing immediately or shortly after the progress bar was fully filled. They believed that it was the *"desired amount" [S8]* when the bar was full:

- *"Originally I had 4 sentences to write. After writing 2 the progress bar is full and it told me the reflection is great, so I stopped right there." [S13]*

- *"the system said that the reflections were good enough" [S28].*

This suggested that we need to be careful when using conclusive feedback, e.g., fully filled progress bars, textual hints such as "great reflection", etc.

5.4.3.3 Quantitative results on latent quality feedback (LF)

In **LF** condition, when participants clicked the *"submit"* button, they saw the system feedback and were able to choose to revise the reflection. Therefore, we attribute the reflection quality improvement and length increase (compared with **NF**) to participants' revisions after they saw the feedback.

We first compare the system-predicted reflection quality between the first drafts and the final submissions in **LF** condition. The average quality of the first draft is 3.11 (σ =1.02). In comparison, the average quality of the submitted draft is 3.78 (σ=0.48). In total they viewed the latent feedback panel for 174 times, among which they chose to go back and revise the reflection for 54 (31.0%) times. Figure 27 shows the participants reactions (i.e. go back to revise, go to next/submit without revision) when they saw the system feedback. Among the 54 revisions, 49 (90.7%) revisions lead to better reflections (Table 5).

Table 5. The distribution of system-predicted quality changes after revision.

		After			
		None	Vague	General	Specific
Before	None	0	**1.85%**	**7.41%**	**7.41%**
	Vague	0	1.85%	**20.37%**	**18.52%**
	General	0	0	7.41%	**35.19%**
	Specific	0	0	0	0

5.4.3.4 Qualitative results on latent quality feedback (LF)

There were 29 occasions when participants chose to submit reflection even though they did not get the *"perfect reflection"* feedback from the system. Reasons include:

- *"I don't think I can write more when I go back." [S21]*

- *"I think I've provided enough details, even though CourseMIRROR still asked me to provide more detail." [S33]*

- *"I was very confused about the '[no] free lunch theorem' and I knew nothing about it so I cannot further illustrate why it is confusing." [S35]*
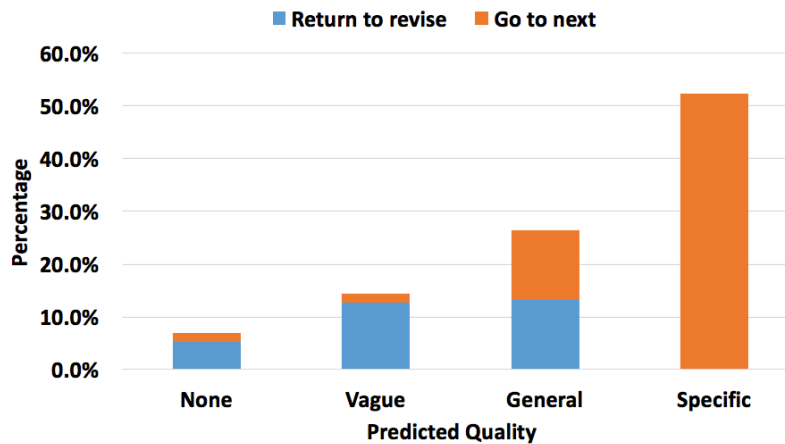
Figure 27. Participants' reactions (i.e. return to revise, go to next) when showing the reflection quality in LF.

5.4.3.5 Tradeoffs between IF and LF

Although the quantitative analysis did not show a significant difference between **IF** and **LF**, we discovered qualitative differences through observations and interviews with participants.

Real-time vs. Attention. Participants expressed their preference on **IF** for its visibility in real-time. At the same time, they reported that they mainly focused on the quality feedback (i.e. the progress bar). Two participants reported that they totally ignored the improvement suggestions and another participant only followed the improvement suggestions when he *"tried hard but still can't fill the bar" [S3]*. By comparison, participants in **LF** reported that they paid sufficient attention to both the progress bar and the textual suggestions. We attribute this to the *intrusive* nature of latent feedback (via a dialogue box), which drew more attention by pausing the composition process.

However, the **LF** can frustrate participants for delayed information:

- *"The system should tell me what is the expected reflection at the beginning rather than after I spend time thinking and writing the reflection." [S40]*

5.4.3.6 Pattern matching improves the accuracy of quality prediction

In order to assess the efficacy of *pattern matching* in improving the quality prediction accuracy, we conducted an off-line comparison between using the classifier only and using the combinations of the classifier and pattern matching (Table 4). The gold standard quality scores were human annotations.

The classifier (i.e. SVM) used in the study was trained on previous student reflection datasets [117] containing 1,257 reflections and the experts' quality ratings. It is worth mentioning that the domain of the course (i.e. data modeling) in this study is different with the domain of the training course (i.e. material science and engineering).

78

Table 6. Accuracies of quality prediction algorithms.

Method	Percent	kappa	QWKappa
Classifier Only	58.3%	0.28	0.67
Classifier + All Pattern Matching	**77.2%**	**0.52**	**0.83**
Classifier + Domain Word List	73.3%	0.46	0.76
Classifier + Quality Patterns	71.7%	0.44	0.80
Classifier + Categorical Patterns	58.9%	0.30	0.70

On average the *domain word matching*, *quality pattern matching*, and *categorical pattern matching* are triggered by 1.12 (σ=0.89), 1.41 (σ=1.06), 0.05 (σ=0.2) times, respectively, for each reflection. The results in Table 6 confirms that integrating pattern matching could enhance the quality prediction accuracy, and mediate the domain miss-match problem.

5.5   IN THE WILD DEPLOYMENTS

CourseMIRROR has been deployed in eight courses[9] in two universities as of September 2016, involving a total of six instructors and 317 students. Most of the courses were undergraduate level STEM courses, such as Basic Physics, Data Structures, and Statistics for Industrial Engineering.

Overall, students reported positive experiences with CourseMIRROR (Figure 28). Ratings were measured on a 5-point Likert scale (1 = *strongly disagree*, 5 = *strongly agree*). Students thought CourseMIRROR was easy to use μ=4.30 (σ =0.80) and would like to use CourseMIRROR in future courses μ=3.96 (σ=0.94).

---

[9] Fan et al [58] is a non–archival publication reporting preliminary findings from two pilot deployments.
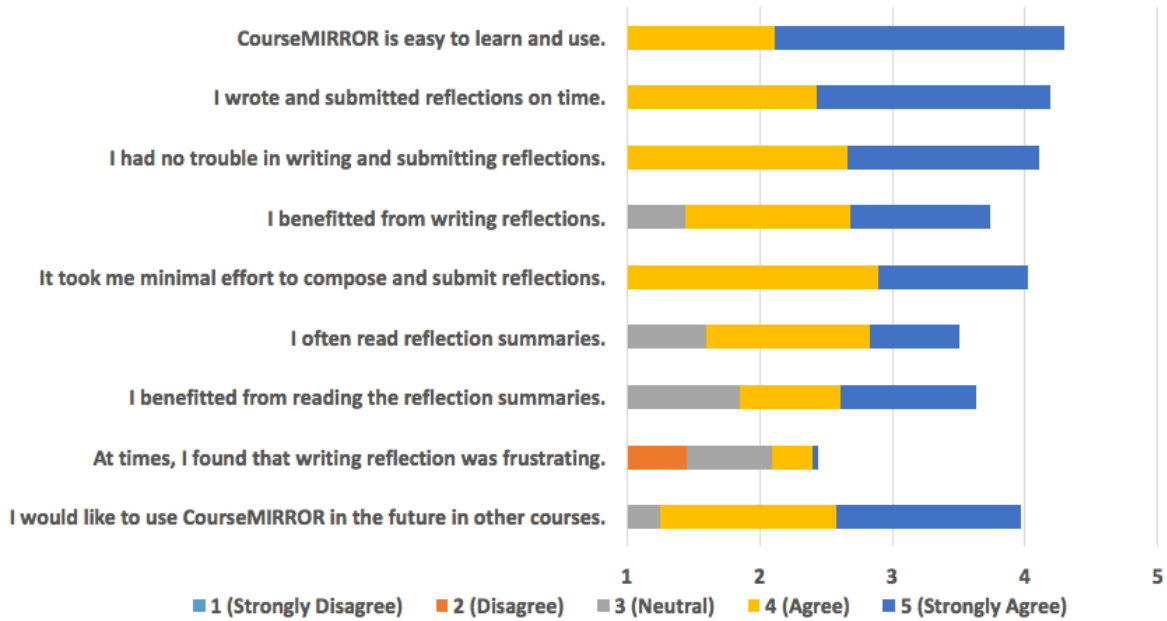
Figure 28. Subjective ratings on a 5-point Likert scale.
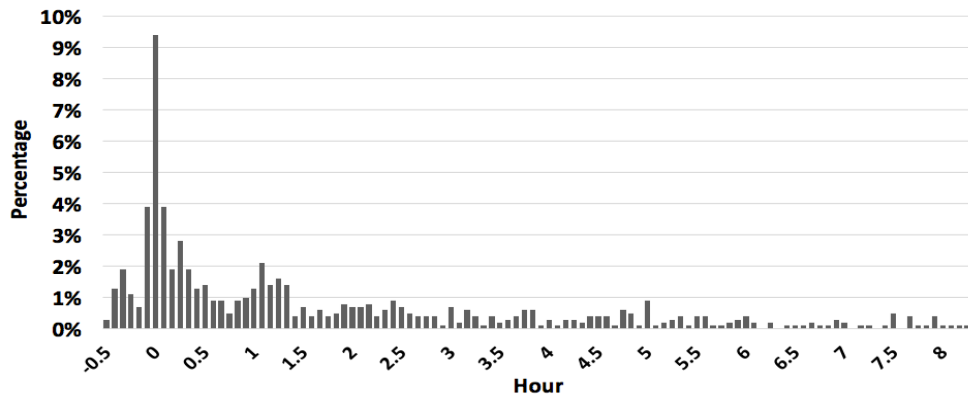
Figure 29. The histogram of response time (hour).

Finding 1: *Students were willing to submit reflections in a timely manner.*

In total we collected 3,855 reflections from the eight deployments. The average response rate was 53.1% (σ =0.16). This rate is encouraging considering that there was a significant portion of quiet and shy students who rarely asked questions or seek for help actively in each lecture.

We further analyzed the submission time of the reflections. We found 48.3% of the reflections were submitted within two hours after the end of each lecture. 9.2% of the reflections were submitted before the end of the lectures (Figure 29). These results confirmed that the

reflections were indeed *in-situ*. We attribute the timely reflection submission in part to the novelty and efficacy of our CourseMIRROR mobile client.

Finding 2: *Students benefitted from the reflection and feedback cycle enabled by CourseMIRROR.*

The average response to question *"I benefitted from writing reflections"* was 3.74 ($\sigma =1.08$) (Figure 28). Students reported that the benefits were two fold. Firstly, composing reflections in CourseMIRROR enhanced their retention by encouraging them to revisit what they learned:

- *"It's not a long time to learn which subjects aren't understood by the class."*
- *"I can think about what I learned and what I didn't understand."*

Secondly, the timely instructor feedback enabled by CourseMIRROR helped students clear up their confusions:

- *"Because our prof used those reflections and cleared the muddy points."*
- *"Especially, when our instructor started to solve more examples on class, I saw this benefit in a more concrete way."*

Finding 3: *Reflection summaries allowed instructors to understand students' difficulties efficiently.*

All the instructors responded positively to CourseMIRROR according to post-study questionnaires and interviews. Instructors reported that they regularly read the reflection summaries generated by CourseMIRROR, e.g., one instructor reported that she *"never skip the summary"* while another instructor reported that he *"tried to look at every summary"*. The time needed to understand the summary for each lecture was minimal, ranging from *"definitely less than 5 minutes"* to *"5-10 minutes"*. In comparison, an instructor spent 30-45 minutes summarizing the responses from a 50-student class in traditional paper-based deployments [123]. The automatic text summarization was promising— e.g., instructors can *"get an idea of the issues some students*

are having trouble with" by reading the summaries, and *"clarify/go over some topics that indicated as problematic"* in future lectures.

Finding 4: *Students enjoyed reading summaries of reflections from their classmates.*

The average subjective ratings of *"I often read reflection summaries"* and *"I benefitted from reading reflection summaries"* on a 5-Likert scale (Figure 28) are 3.51 (σ =1.09) and 3.63 (σ =1.06), respectively. They reported that seeing their classmates' reflections could broaden their views and allow them to reevaluate from different perspectives (e.g., *"I feel I was also confused about other people's muddiest points after I see the summary"*). At the same time, realizing that other students having the same confusion could reduce their frustration and enhance their confidence (e.g., *"Good to see other people were also confused, I know it's not my problem and relaxed"*).

Finding 5: *The quality feedback feature can help students compose higher-quality reflections in real-world settings.*

After finishing the lab study on the refection quality feedback feature, we integrated the updated CourseMIRROR client with instant quality feedback to the Data Structures course (29 lectures in total, 40 CS undergraduate students enrolled) in a local university in Spring 2016. The feature was enabled in an app update made in the middle of the semester. 12 students updated the app (2 started from lecture 19, the other 10 started from lecture 20). The following analysis focuses on the reflections generated by the 12 students who used both versions (i.e. with/without quality feedback). Specifically, we compare their reflections submitted from lecture 9-18 (i.e. without quality feedback) and lecture 20-29 (with quality feedback) to see whether the feedback could help students generate more specific reflections.

To measure reflection quality, the same two raters (who rated the reflection corpus in the lab study) rated the reflections collected from the 12 students with the same rubrics ([117], Figure 21). Their independent ratings achieved high agreement (percent agreement: 88.5%; Cohen's kappa: 0.73; Quadratic Weighted Kappa: 0.93). They discussed on the disagreements to achieve consensus.

Table 7 shows the comparison of the reflections generated by the 12 students with/without interactive quality feedback. Students composed significantly longer (12.5 vs. 7.5, $p<0.001$) reflections with interactive feedback. At the same time, the reflection quality was also significantly higher (3.4 vs. 2.8, $p<0.05$) with interactive feedback. Considering the quality feedback feature was updated in the middle of the semester, this is not a tightly controlled comparative study. However, considering that the reflection length and quality *decreased* over time in previous studies without the quality feedback feature, this result is still promising. It implied that the interactive feedback motivated students write higher-quality reflections in a sustainable manner. We plan to conduct larger scale and controlled deployment in the future to verify this finding.

Table 7. Distribution of reflection quality in the deployment with and without quality feedback.

	W/O Interactive Feedback	W/ Interactive Feedback
Total # of Reflections	86	79
Average Length	7.5	12.5
Reflection Quality	Average: 2.8	Average: 3.4
None(1)	21 (24.4%)	6 (7.6%)
Vague(2)	7 (8.1%)	5 (6.3%)
General(3)	27 (31.4%)	19 (24.1%)
Specific(4)	31 (36.0%)	49 (62.0%)

Students reported positive experiences with the interactive quality feedback. On a 5-Likert scale, they reported that the interactive feedback and suggestions were relevant to their reflections (μ=4.29, σ =0.82). They also reported that the interactive feedback helped them think deeper and compose more specific reflections (μ = 4.6, σ = 0.68). Sample comments include:

- *"The new app was helpful most of the time, especially when I only gave a general idea, it pushed me to think deeper about what I'm interested or confused, and be able to find the specific point."*

- *"I really think more carefully about the lesson when writing reflections using the updated version."*

Students also reported that the interactive feedback helped them learn how to do deep reflection and the ability could last:

- *"After the first 1-2 times, I knew what is a desired reflection and I can write a 'perfect reflection' without reading the suggestions then."*

Although the overall reflection quality improved, there were still 7.6% non-substantive reflections submitted. The major complaint on the interactive feedback feature was the lack of diversity. One student reported that *"the pattern of the suggestions seems to be fixed"*. In the future, we plan to significantly increase the diversity of the feedback to avoid boredom, e.g., by changing the presentation or by integrating more pattern matching templates to make the feedback more specific to the input.

Students also hoped that CourseMIRROR can provide improvement suggestions not only relevant to the input, but also relevant to the lecture content not mentioned in the current reflection. One student wished that the system could *"list some topics in the lecture"* so that he can *"pick up the most confusing point and explain"*.

We also discovered minor gaming behaviors by analyzing the user interaction log. For example, one student originally wrote *"N/A"* and got the quality feedback as "none" reflection. After that, the student tried to get a higher score by rephrasing the reflection, such as *"no muddy point"*, *"all clear and no muddy point"*, and finally submitted as *"everything is confusing"*. In the future, we need to detect such gaming behavior in real time and provide scaffolds explicitly designed for gaming behaviors, e.g., by prompting the student to explain the *why* behind a concept mentioned in the lecture.

Finding 6: *Active integration to the curriculum is essential.*

Students were not mandatorily required to participate in any of the deployments. We explicitly notify students who opted-in that they were free to quit the deployment at any time. Although we observed high response rates in most deployments, we cannot claim that CourseMIRROR could always work in every condition. For example, the response rate (24.8%) in the Basic Physics class in Spring 2015 was significantly lower than other deployments (e.g., 56.7% in Statistics for Industrial Engineers, 57.7% in Mobile Interface Design). We attribute the low response rate to the weak integration to the course curriculum.

First, there were no course incentives (i.e. extra credit) provided in this deployment. Surprisingly, according to past experiences in deployments, course incentives (as low as one extra point in class participations) worked better than monetary incentive (e.g., as much as $30 for semester-long participation). Thus we encouraged, but did not require, instructors to provide some extra credit for participation in later deployments.

Second, the instructor did not refer to CourseMIRROR in class after he announced the deployment in the first lecture. We found that it was more effective for the instructor to explicit

acknowledge the source of clarifications, i.e. CourseMIRROR, in the reflection and feedback cycle.

5.6   LIMITATIONS AND FUTURE WORK

We have not completed a controlled study in classroom deployments to compare the learning outcome for two reasons. First, CourseMIRROR was evolving through the iterative design and deployment process (e.g. the iOS version was available two terms after the Android version. The instant reflection quality feedback feature was added based on lessons learned from earlier deployments). The dynamic nature of CourseMIRROR made it hard to freeze all the features and run a controlled deployment at the early stage. Second, we need a course with high opt-in rate (ideally 40 or more students per condition) to get the statistical power needed to analyze the learning outcome in a semester-long deployment.

Although both lab studies and in-the-wild deployments show the benefits of CourseMIRROR in facilitating and scaling reflection prompts, it is still necessary to improve and deploy CourseMIRROR in even larger scale, more diversified courses in the near future. More importantly, we plan to conduct large scale class deployment with *control groups* to further verify the educational value of CourseMIRROR in different contexts (e.g. What would be the best practices for deploying CourseMIRROR? Whether and to what extent CourseMIRROR combine with other instructional interventions synergistically?).

Another interesting future work is to enable *personalized learning* by analyzing reflections collected via collaborative filtering algorithms. Potential opportunities include recommending

relevant learning materials (e.g., MOOC videos) and exercises, as well as establishing the connection and collaboration among peers with complementary skills.

While reading the summaries generated by CourseMIRROR can help instructors understand students' difficulties and misconceptions, there still exists opportunities to facilitate instructors to convert summaries to *actions* and *resources* in the follow-up lectures. We plan to explore techniques (e.g. instructor-side visualizations, revision tracking, and improvement suggestions) to scaffold instructors to carter the upcoming teaching activities according to reflections from students.

5.7   SUMMARY

We presented the iterative design, prototype, and evaluation of CourseMIRROR, a mobile learning system that uses NLP techniques to enhance large classroom instructor-student interactions via streamlined and scaffolded *reflection prompts*. CourseMIRROR reminds students to compose their reflections directly on their mobile devices *in-situ* after each lecture. CourseMIRROR also scaffolds students to compose high quality reflections and facilitates both instructors and students to identify major points of confusion in a lecture via customized natural language processing algorithms. We conducted both controlled lab studies and eight semester-long deployments to evaluate the efficacy of CourseMIRROR. Overall we show that the reflection and feedback cycle enabled by CourseMIRROR is scalable and beneficial to both instructors and students. CourseMIRROR is freely available for classroom usage at http://www.coursemirror.com.

6.0   TONEWARS: MASTERY LEARNING OF SECOND LANGUAGE THROUGH ASYNCHRONOUS MODELING OF NATIVE SPEAKERS IN A COLLABORATIVE MOBILE GAME

"Those who know nothing of foreign languages, knows nothing of their own."

— *Johann Wolfgang von Goethe*

6.1   BACKGROUND AND MOTIVATION

Mandarin Chinese, the world's most widely spoken language, has grown in popularity as a second language nowadays. In 2010, about 750,000 people took the Official Chinese Proficiency Test [106]. In 2015, 200,000 U.S. students were studying Mandarin; President Obama hopes to see the number quintuple to 1 million by 2020 [61].

There are several unique challenges for English speakers when learning Mandarin Chinese as a Second Language (CSL). The learners need to get familiar with the *logographic* writing system (compared with the *alphabetic* writing system of English), memorize around three thousand characters, and become familiar with the tonal sound system. Among all these challenges, learning Chinese tones is often considered as the most difficult task [96, 151]. The tones in Chinese determine *meaning* while tones in English are used for *grammatical and expressive inflection*. Multiple cross-linguistic studies [103, 166] suggest that linguistic experience

plays an important role in tone perception and the source of difficulty in learning tones has generally been attributed to the interference from English stress and intonation systems for American students [166]. English listeners may perceive Mandarin high tones as stressed and low tones as unstressed; however, in Mandarin, the stress is realized more by the duration and amplitude than tonal changes [164].

One common paradigm for effective second language learning is to delve into a native speaker environment [105, 108]. This idea has been leveraged by ToneWars [76] for engaging language learning recently. ToneWars connects CSL learners with native speakers in a collaborative mobile game. CSL Learners can practice tone recall, perception and production by directly competing with native speakers in ToneWars. Head et al [76] observed an average gain of 6.2 tones in short term recall after 40-minute gameplay.
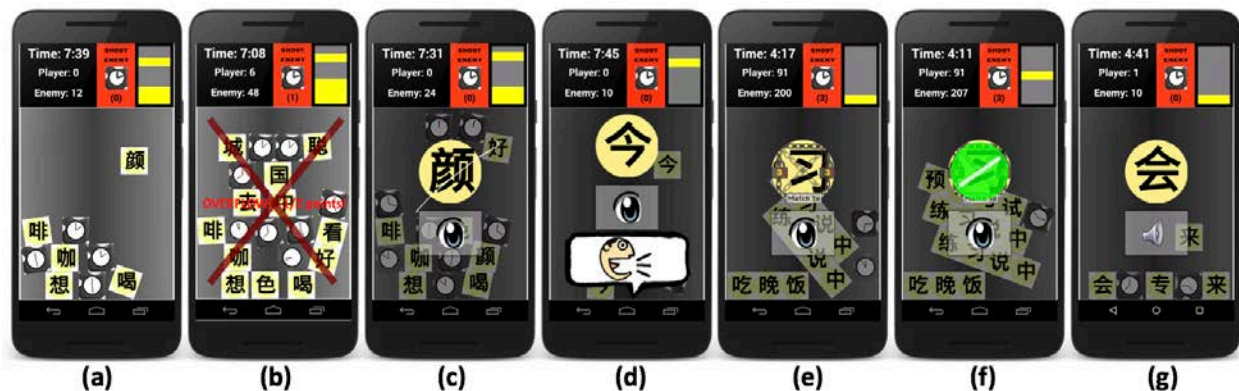
Figure 30. ToneWars Screenshots. (a) Phrases fall and collide; (b) The phrase stack overflows and clears, the player loses points; (c) The player traces a tone with a touch gesture to eliminate the character; (d) The player uses speech to input a tone; (e) A character locks after an incorrect guess; (f) Visual hint for a locked character; (g) The player listens to the audio hint by clicking the speaker button.

While ToneWars provides new insights on including native speakers to assist second language acquisition in a collaborative learning game, the original study leaves a number of important questions unanswered. First, the synchronous nature of game play requires the same

number of native speakers and CSL learners in each learning session. Although native speakers reported compellingly favorable experiences in the lab study [76], there may not be sufficient native speakers to connect and compete with CSL learners due to time zone difference and the time constraint of native speakers. Second, given the drastic skill differences between learners and native speakers, how can learners build and maintain their *self-confidence* in the competitions with native speakers? Since self-confidence is a basic determinant of learners' motivation in second language learning, we should guarantee that learners can gain their confidence, rather than being frustrated in the competition. Third, while the lab study showed promising results for improving learners' *short-term recall*, can ToneWars provide *measurable improvement* in learning outcome in a longitudinal study?

In this chapter, we extend the prior work in three ways. First, we address the scalability issue via asynchronous modeling of native speakers. Specifically, we model the *interaction patterns* of native speakers for offline competition. By recording the key interaction patterns, learners are able to play against the pre-recorded experts at any time, regardless of native speaker availability. We also model native speakers' *language skills* in fine grain (i.e. phrase level tone recall) and use it as the goal for learners to achieve *"bite-sized"* native level mastery. Second, based on the modeling of native speakers' language skills, we propose a metric *"Native Level Index"* (**NLI**) to measure whether a CSL learner achieves native level mastery for a specific phrase at a given moment, as a quantitative indicator of a learner's performance. We believe such fine-grained modeling and feedback on language mastery can enhance learners' self-confidence thus can motivate them in a sustainable manner. Last, we conducted a 3-week study with 18 CSL learners to investigate the effectiveness of such modeling in improving learning and maintaining learners' motivation. We found that asynchronous gameplay can significantly facilitate learning,

without losing user satisfaction. Participants had an averaged absolute learning gain of 29.7 and a relative gain of 64.4% for tone acquisition over the three weeks. We also found that CSL learners can achieve native level proficiency in 58.2 out of 69 phrases at the end of the study. All participants reported positive and favorable experiences with ToneWars.

Our research provides the following implications for researchers building mobile learning technology to aid second language acquisition beyond learning Chinese:

1) We demonstrate the feasibility of leveraging native speakers as both a *benchmark* for language mastery and a *motivator* for language learning. We believe that this direction could lead to interesting future research.

2) We propose a *scalable* approach to enable authentic competition and skill comparison with native speakers via modeling both the *interaction patterns* and *language skills* of native speakers asynchronously, and prove the effectiveness of such modeling in a longitudinal study.

3) We show the feasibility and efficacy of maintaining learners' sustained motivation through *fine-grained* skill modeling, feedback, and comparison with native speakers in the context of a competitive mobile game.

The *intelligence* of ToneWars involves modeling both the interaction patterns and language skills of native speakers and enabling one-on-one tutoring experience for language learners. The content of this chapter can be found in the published paper [60].

6.2   TONEWARS IN ACTION

To our knowledge, ToneWars (Figure 30) is the first language learning system connecting second language learners and native speakers via collaborative mobile gameplay. Learners practice

Chinese tones via direct competition with native speakers in synchronous gameplay. ToneWars integrates gameplay elements from popular mobile games such as Tetris and Fruit Ninja to provide an engaging experience for both native speakers and learners. In this section we present a scenario to demonstrate how ToneWars works in action.

Claire is a first-year, second language learner of Mandarin. On a study break she finds ToneWars, a game for practicing tones for phrases she learned in class. After connecting to a native speaker, she slashes Chinese phrases with the shapes of the tones of the characters (Figure 30.c). She can also pronounce the character into the microphone (Figure 30.d). As she does this, she eliminates the phrases and keeps her screen from overflowing (Figure 30.b), which will penalize her by halving her score. With each match, Claire earns points and saves a block. By tapping the "shoot enemy" button, she drops blocks to clutter the stack of her opponent. She monitors how cluttered her opponent's screen is through a preview pane in the top-right corner of the screen. Her score tells her when she is matching tones more successfully than the native speaker, which motivates her to continue playing and practicing.

When Claire makes a mistake, the character will be locked (Figure 30.e). To unlock and proceed, Claire clicks the hint button to view the *visual* hint (Figure 30.f) which shows the shape of the correct tone of that character. She can also listen to the *audio* hint (Figure 30.g) which is the spoken form of the character pre-recorded by a native speaker. After learning the correct tone, Claire enters the correct tone 4 times and the character becomes unlocked. This process emulates language learning drill exercises.

Claire may also play against a pre-recorded native speaker when there is no native speaker available. With asynchronous gameplay and offline performance competition, Claire experiences engagement and personal satisfaction in her improvement and success just like before.

6.3   RELATED WORK

6.3.1   Mandarin Tone Learning

Despite the challenges of tone acquisition [96, 151, 164, 166], research has shown that English speakers can be trained to successfully differentiate between Mandarin tones [104, 163], as well as to produce Mandarin tones [164]. Researchers have explored leveraging both *perception* training (e.g., by listening [163]) and *production* training (e.g., by speaking [104]) for tone acquisition. Wang et al. [163] showed that, after *perceptual* (i.e. auditory) training of Mandarin tones, the American learners' ability to identify tones improved significantly (21% improvement) and the improvement could be generalized to new stimuli and retained for six months. For *production* training, Leather [104] found that Dutch speakers were able to perceive the differences in tone after they were trained to produce Mandarin tones and concluded that training in one modality tended to be sufficient to enable learners to perform in the other. ToneWars provides CSL learners with both *perception* training and *production* training. Auditory training is provided to learners when they listen to audio hints (Figure 30.g) during gameplay. This emulates the multi-modal learning experience in Chinese classrooms, where students listen to tones spoken by their instructor and trace their shape in the air. The speech input (Figure 30.d) mode was designed to help learners' transition from declarative knowledge, where they can recognize a word, to productive knowledge, where they can use it [14].

Researchers also explored the idea of using color encoding [50, 64] and gestures [64, 122] to aid tone acquisition. The pitch contours make tones conducive to visual depiction [122]. Moreover, adding body motion information (e.g., gestures) could create memory traces that are even more multi-model and increase the learning robustness [122]. In real world Mandarin

classrooms, gestures illustrating the spatial metaphor of pitches are widely adopted by instructors and students [64, 76], not only because they can reinforce learning, but also because they can make the classroom more engaging [76]. However, the social acceptability problem [141] may prevent CSL learners from tracing tones when practicing in public. Inspired by previous research, ToneWars enables the touch gesture input method (Figure 30.c) during gameplay. We believe that learners could benefit from the same audio-kinesthetic association as in classroom with input that is more socially acceptable [141] outside the classroom.

6.3.2 Native Speakers in Second Language Education

In recent years, a great deal of research and pedagogical experimentation has been conducted to investigate how to effectively leverage native speakers in second language education [90, 105, 108, 156, 164]. One important recommendation is to allow native speakers to play the role of an expert assisting the learner in improving both linguistic and cognitive skills related to the language. Native speakers can offer authentic language discourse to help language learners acquire new lexical items and correct grammatical structures [105, 108]. Moreover, the experience could help learners gain confidence and motivate them to engage in conversations in the future [76, 105]. However, there are two major limitations. First, native speakers are usually inaccessible to learners. Second, according to *"the input hypothesis"* [98], language acquisition can only occur when the input is comprehensible for the learner. In other words, language heard but not understood is of little or no use for learning purpose. However, in practice, native speakers may use some language that is beyond the comprehension level of the learners [76, 105]. Such perceptions of inequality may even lead to a lack of confidence and anxiety [105] towards further learning.

Beside the role of assisting language learners directly, native speakers' performance can also be treated as the "*native norm*" [164] for assessment purposes. For example, Wang et al. [164] normalized the pitch contours of Mandarin tones among native speakers and used the normalized *F*0 as the norm when evaluating learners' tone production.

Inspired by these research, ToneWars enables the interaction between CSL learners and native speakers in gameplay. The direct competition not only makes the game more fun, but also serves as a motivator for CSL learners, especially when learners know that they can perform as well as native speakers. By asynchronous modeling of native speakers, ToneWars is highly scalable so that learners can play against native speakers at any time.

6.3.3   Mobile Language Learning Systems

The mobile phone is a great platform to implement *anywhere*, *anytime* micro-learning opportunities, since it always accompanies its owner wherever she may go. There is a great deal of mobile language learning systems [1, 12, 29, 47, 54, 53, 52, 88, 100, 157] with which learners can leverage the brief fragments of free time that spaced throughout the day for language learning tasks. A number of mobile systems have been proposed to enable context-aware learning experience, e.g., by providing learners with learning content that is relevant to their locations [1, 47, 53], objects they are interacting with [12], as well as their learning histories [52]. Moreover, these systems utilize the unique capabilities of mobile phones (e.g., SMS [29], speech recognition [100], multimedia [12, 54, 88, 100, 157]) to make the learning experience more multi-modal and more fun.

A number of mobile applications [1, 54, 53, 52, 157] have been developed to address the challenges of learning Mandarin Chinese. Some of them focus on vocabulary learning in general

[1, 53, 52], others focus on the tonal sound system [54] and the logographic writing system [157]. They were designed for either CSL learners [1, 54, 53, 52] or native speakers in elementary schools [157].

MicroMandarin [53] provides learners with vocabularies that are relevant to their locations (e.g., suggest "Cappuccino" in a Cafe). MemReflex [52] considers a broader sense of context-aware learning—adaptive to personal learning history. ToneWars is different with these two systems in that it is an educational game that provides learners both engaging gameplay and effective learning experiences. In addition, ToneWars specifically focuses on tone acquisition, while the other two systems focus on vocabulary learning in general.

The Multimedia Word and Drumming Strokes mobile games [157] were inspired by traditional Chinese group games and aimed to improve native speakers' language abilities. Players sit together and share a mobile phone in the gameplay. In comparison, ToneWars connects CSL learners with native speakers who are separated physically and aims to improve CSL learners' language abilities.

Tip Tap Tones [54] is a mobile game with the purpose of training CSL learners to acquire the tonal sound system. Tip Tap Tones provides aural tone perception training via single-player flashcard-style drills at single-character level. In comparison, ToneWars supports phrase-level tone learning and connects CSL learners with native speakers in multiplayer gameplay. In addition, ToneWars supports multi-modal interactions (e.g., gesture, speech) inspired by tone exercises in Mandarin classrooms.

6.4   ASYNCHRONOUS MODELING OF NATIVE SPEAKERS

As we discussed earlier, the scalability issue is the main challenge for ToneWars when deployed in the wild. Although native speakers reported favorable and engaging gameplay experience in previous lab study [76], it is likely that with time zone differences, there might not be enough native speakers to play with CSL learners online at the same time. Therefore, we integrate asynchronous gameplay to ToneWars, so that learners are able to play and practice at any time, regardless of native speaker availability.

The role of native speakers in ToneWars is twofold: 1) competing with learners directly in gameplay to make the game itself engaging and fun; 2) serving as a motivator for language learning—learners would be highly motivated when they know that they can perform as well as native speakers, even only on a subset of the vocabularies.

Therefore, the asynchronous competition should achieve two goals: 1) delivering authentic and engaging gameplay experiences for CSL learners; 2) allowing offline language skill comparison between learners and native speakers.

We achieve these two goals via asynchronous modeling of both *interaction patterns* and *language skills* of native speakers. By recording the key interaction patterns, learners are able to play against the pre-recorded experts at any time and we hypothesize such experience could be as engaging as synchronous gameplay. Moreover, we built an offline model to describe the language skills of native speakers. Therefore, the language skill competition between CSL learners and native speakers can also be conducted anytime.

6.4.1 Interaction Pattern Modeling

In recording mode, ToneWars records native speakers' key interaction patterns along with the corresponding time stamps, including tapping to select a phrase, slashing a gesture to match, pressing the attack button, etc. When native speakers are not available, CSL learners play against the pre-recorded activity scripts. ToneWars plays the activity script in a continuous loop (Figure 31) when the duration of the recording is not sufficient.
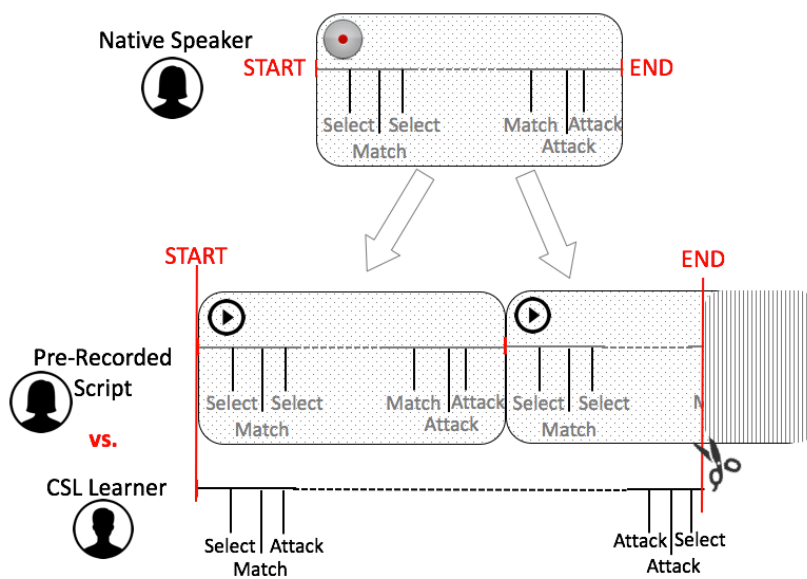
Figure 31. The activity script is played in a continuous loop if the length is not sufficient.

One major challenge for this approach is how to make a game where players felt they were playing in real-time against the recorded opponent. Real-time interaction between the competitors is a key element of making a competitive game engaging and fun. The problem with replaying past sessions against new players is that the real-time interactivity (i.e. action-and-reaction) is lost. For example, prerecorded players will not be able to fire back at their opponents out of 'spite' (i.e. attack right after they have been attacked), or time their attacks when their opponents are doing particularly well—all of their attacks are going to happen at prerecorded times. One compromise is to lower the time sensitivity of attacks; in other words, design the attacks such that they will

98

have the same effect when they are replayed during a new competition. We decided to make the effect of attacks last longer—the blocks will remain active in the phrase stack for 20 seconds. Under such a design, an attack of dropping blocks will always drop blocks on and inconvenience the player, and such an effect will still match the intent of the player from the original game when replayed.

Even though the pre-recorded opponent cannot *react* to player in real-time, they can still *interact* a lot. Beside the attacks with long-lasting effects, the preview pane in the top-right corner allows players to monitor how cluttered the opponent's screen is, which could also enhance the interactivity. We believe that these designs would make the competition still feel rich, even if the pre-recorded opponent does not really respond to player's actions.

6.4.2   Language Skill Modeling

As we mentioned, language learners could be highly motivated when they can achieve native speakers' proficiency. In order to allow CSL learners to compare their language skills with native speakers offline, we build a model to describe the language skills of native speakers.

First of all, we need to find a metric in the game that can accurately measure the language proficiency of the players. We find that the elimination time of a block (i.e. the time needed from tapping the block to finishing the input of the correct tones so that the block is eliminated) is a key indicator of players' language proficiency. When the player is more proficient with the language, she can eliminate blocks faster since she needs less time for cognitive processing and could be able to save the time for recovering from incorrect guesses. Therefore, we use this as the main metric in the model.

To collect native speakers' data for modeling, we invited 7 native speakers to play ToneWars for 25 minutes each and recorded logs during the gameplay. Then we discarded the first 5 minutes, during which they got familiar with the game design and control mechanisms. During the gameplay, they eliminated phrases (lengths ranged from 1 to 5) combined by 100 Chinese characters[10]. In the paper based test before the data collection, all of them achieved 100% accuracy in tone recall on this vocabulary set. Therefore, we believe this performance dataset can represent native level proficiency.

Originally we planned to use *Character Elimination Time* (**CET**) as the main metric, which is the time needed to eliminate one character. Later on we found that this metric is limited in that it is based on the assumption that the player needs the same amount of time to eliminate one certain character, no matter it appears alone (e.g., a single character '希'), or it appears in a multi-character phrase (e.g., the character '希' in the phrase '希望'). However, appearing in a multi-character phrase could give the player more context which may affect their response time. We did further analysis on this. We found that the distributions of a character's CET differ when it appears in a phrase with different lengths. For example, the mean CET for single character is longer than the other four conditions (i.e. when a character appears in a multi-character phrase). This suggests that CET may depend on whether the character appears alone or appears in a phrase.

Therefore, we propose to use *Phrase Elimination Time (**PET**)* as the main metric, which is the time between the player tapping a phrase to select, and the phrase getting eliminated once the correct gesture was detected. In this case we treated the phrase as a whole, and the model considers the effect of context information included in phrases. We divided the phrases into five groups

---

[10] The vocabulary set was the same with the one used in the 3-week study presented later.

based on their length (i.e. from 1 to 5). For phrases with the same length, we got a list of *PET* from the 7 native speakers. Figure 32 shows an example of the distribution of *PET*s for phrases with "L=1". Table 8 shows the mean and standard deviation of each Gaussian distribution. These models demonstrate the native speakers' performance with ToneWars.

Table 8. Parameters of Gaussian distributions describing native-level performance.

Phrase Length	μ (ms)	σ	Native-Level Index (μ+σ)
1	954.14	363.38	<= 1317.52
2	1848.19	537.07	<= 2385.26
3	2709.71	494.84	<= 3204.55
4	3353.46	558.41	<= 3911.87
5	4278.91	981.85	<= 5260.76

To determine whether a CSL learner achieves native-level proficiency, we need to set up a decision bound. Here we use "μ+σ" as the "*Native-Level Index*", which is the decision bound of whether the CSL learner achieves native-level proficiency. Generally speaking, if a CSL learner can achieve "Native-Level Index" on a certain phrase, he/she performs better than at least ~16% of the native speaker players on that phrase (Figure 32).
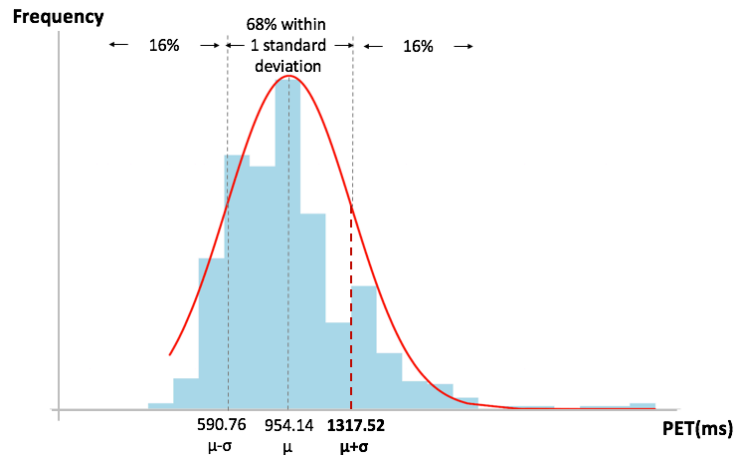
Figure 32. Native-level proficiency with a Gaussian distribution model.

101

6.5   EVALUATION

We conducted a 3-week study with 18 CSL learners to investigate the following three questions:

- Can ToneWars (asynchronous gameplay) help CSL learners achieve mastery on Chinese tones in a relatively long term?

- What are the strengths and limitations of different feedback types (visual vs. audio) and how they affect learning?

- Is the asynchronous gameplay engaging enough to sustain CSL learners' motivation?

In this study, we required participants to visit our lab once or twice (based on their schedules) per week for 3 weeks. During their visits, they were asked to play ToneWars for 1 hour per week, in total 3 hours. They were exposed to both visual feedback and audio feedback during the study. They were told explicitly that the native speaker opponent in the game was rebuilt from a native speaker's activity logs. We conducted pre-test and post-test to measure the learning gains. All the user activity logs were recorded during the game play. We also conducted semi-structured interviews to solicit their subjective feedback.

6.5.1   Participants and Apparatus

We recruited 20 participants for the study from two local universities. 2 of them quitted the study after the first week because of their tight schedules. The other 18 participants completed the study and we only report the results gathered from them. All of the participants were native English speakers actively learning Mandarin Chinese. Self-reported Chinese learning experiences were distributed as: less than 1 year=3, 1-2 years=10, 2-3 years=4, and 3-4years=1. Beside Mandarin, 6

of them also expressed experiences in learning other languages, including: Cantonese, Hindi, Spanish, and Japanese.

Ages of the participants ranged from 18 – 32 (median age 21), with 7 males and 11 females. We compensated participants $30 for their time at the end of the study.

Over the course of 3-week study, participants used a LG Nexus 5 smartphone, which has a 4.95 inch, 1920 x 1080 pixel display, Quad-Core 2.3 GHz Krait 400 Processor, and runs the Android 5.0 OS.

6.5.2   Learning Materials

We worked with a CSL instructor to select 100 characters for the study from *Integrated Chinese Part 1*, a popular CSL textbook in North America. To create the audio feedback, the instructor recorded the pronunciations of the characters which were loaded in ToneWars afterwards. We suggested the instructor speak them slowly and to exaggerate the tone as she might do for novice learners.

6.5.3   Method

We adopted a within-subjects study design based on hint feedback types—visual vs. audio. We first divided the 100 characters into two groups (group A and group B) of roughly equal difficulty. Nine participants used audio feedback version to learn group A and use visual feedback version to learn group B. The other nine participants use audio feedback to learn group B and use visual feedback to learn group A. In each week, each participant spent 1 hour in game play, which was further divided into 12 5-minute sessions (6 audio feedback sessions and 6 visual feedback

sessions, the order was counter balanced). Participants can take a rest any time they wanted between the 5-minute sessions. The order of character appearances was randomized in each session. During gameplay, we collected logs of significant game actions users took, including all match attempts, gestures completed, and interactions between the player and the pre-recorded expert.

We conducted a pre-test at the beginning of the study to establish a baseline. Participants completed a quiz in which they determined tones and pinyin of the 100 characters loaded in ToneWars. Participants completed a post-test which was identical to the pre-test after finishing game play. We did not conduct a test in each week because we did not want the tests themselves to be a factor that could influence the participants' acquisition. After the study, participants contributed feedback through a paper-based survey and a semi-structured interview.

Beside occasional pronunciation feedback in weekly Chinese classes, participants did not engage in any dedicated tone training during the 3-week study. We also ensured that the learning materials loaded in ToneWars do not overlap with their learning materials in their Chinese classes for the duration.

6.6   RESULTS AND DISCUSSIONS

6.6.1   Learning Gain (Overall)

All of the 18 participants improved from their pre-test level of tone recall and pinyin recall. For the pre-test, CSL learners correctly write down the pinyin of 65.6 (min=4, max=98, SD=26.8) characters, among which they correctly recognized the tones of 46.1 (min=2, max=74, SD=22.9)

characters. After 3-week gameplay, they could correct recognize pinyin of 82 (min=8, max=100, SD=26.5) characters, and tones of 75.8 (min=10, max=95, SD=24.0) characters. The average gain of pinyin recognition is 16.4 (min=4, max=29, SD=9.3). The average gain of tone recognition is 29.7 (min=6, max=53, SD=13.7). Pairwise t-tests show that both of these two differences are significant (pinyin: 65.6 vs. 82, $t(17)=7.51$, $p<0.001$; tone: 46.1 vs. 75.8, $t(17)=9.18$, $p<0.001$).
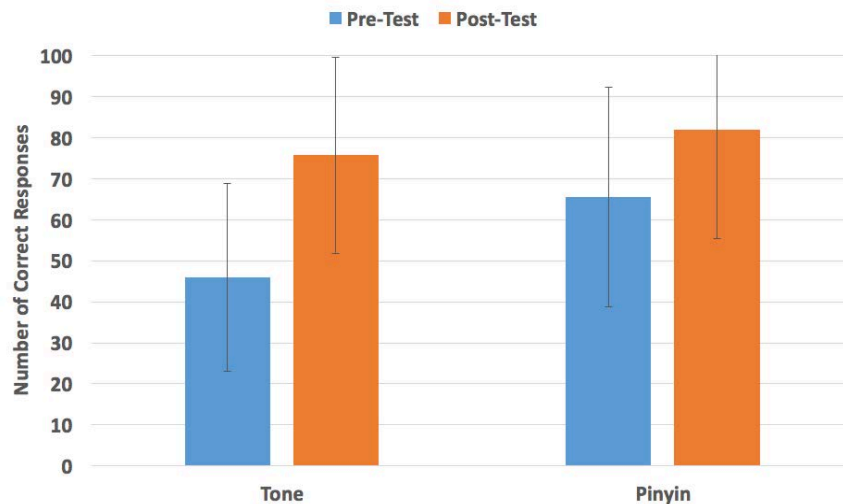
Figure 33. Participants performance of writing the tone and pinyin of 100 characters in pre-test and post-test.

The improvements were also perceived by the participants:

S4: *"I learned quite a bit regarding tones and characters. I definitely saw some improvements."*

S7: *"I learned a lot more tones and it was more helpful to me than I thought it would be. I have understood more Chinese tones and how to say some of them. It has also taught me that the tone markings are important to learning Chinese."*

S13: *"I feel like I have corrected my knowledge of pinyin tone marks for multiple Chinese characters and am now more confident speaking those words because I know how to pronounce them with their correct tones."*

Even though every participant improved, the learning gains varied. Subject 8, who had the smallest gain in both pinyin and tone recall, always exploited hint to preview the answer before making a guess during the gameplay. He found that this was an easy way to *"earn points quickly"* and *"avoid being locked"*. At the same time, based on our observation, this participant did not try to internalize the correct answer for the future; instead, he only tried to earn easy points by following the hints without mental processing and memorization. In comparison, all of the other participants tried to retrieve their impression and made their guess before viewing hints; in very rare conditions they viewed the hints first. S8 also expressed that he had no interest in any kind of mobile games. Even though S8 did not learn as much as the other participants, he still gained 6 tones and 4 pinyin during the study.
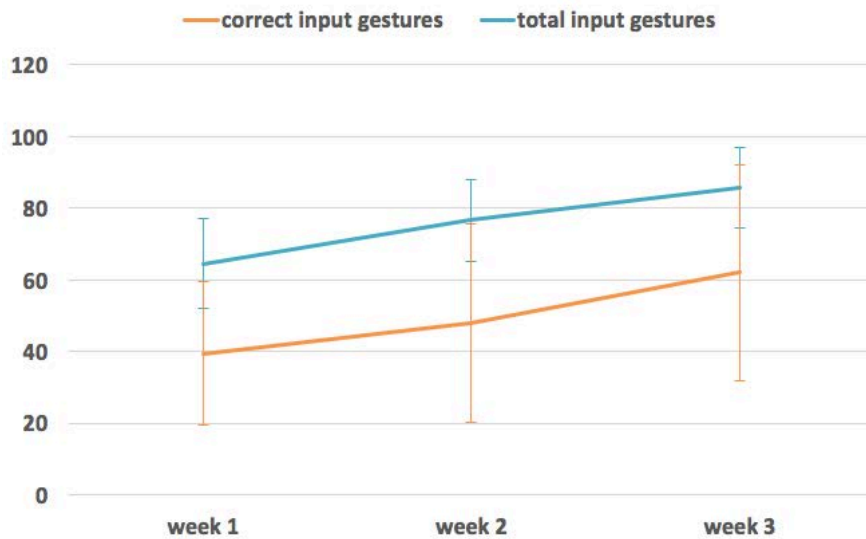
Figure 34. Average number of gestures and correct guesses in a 5-minute session over the course of the 3-week study.

Beside the improvements between pre-tests and post-tests, we also observed that participants in general improved in the number of correct guesses that they could make over time. Figure 34 shows the average number of gestures and the average number of correct guesses the user made during each 5-minute round over the course of 3-week study. There was a significant

increase on the average number of gestures (64.5 vs. 85.8, t(17)=7.28, p<0.001) and number of correct guesses (39.4 vs. 62.0, t(17)=4.37, p=0.001) the user made during each round from week 1 to week 3. These growing numbers also imply participants' mastery of the learning material over time.
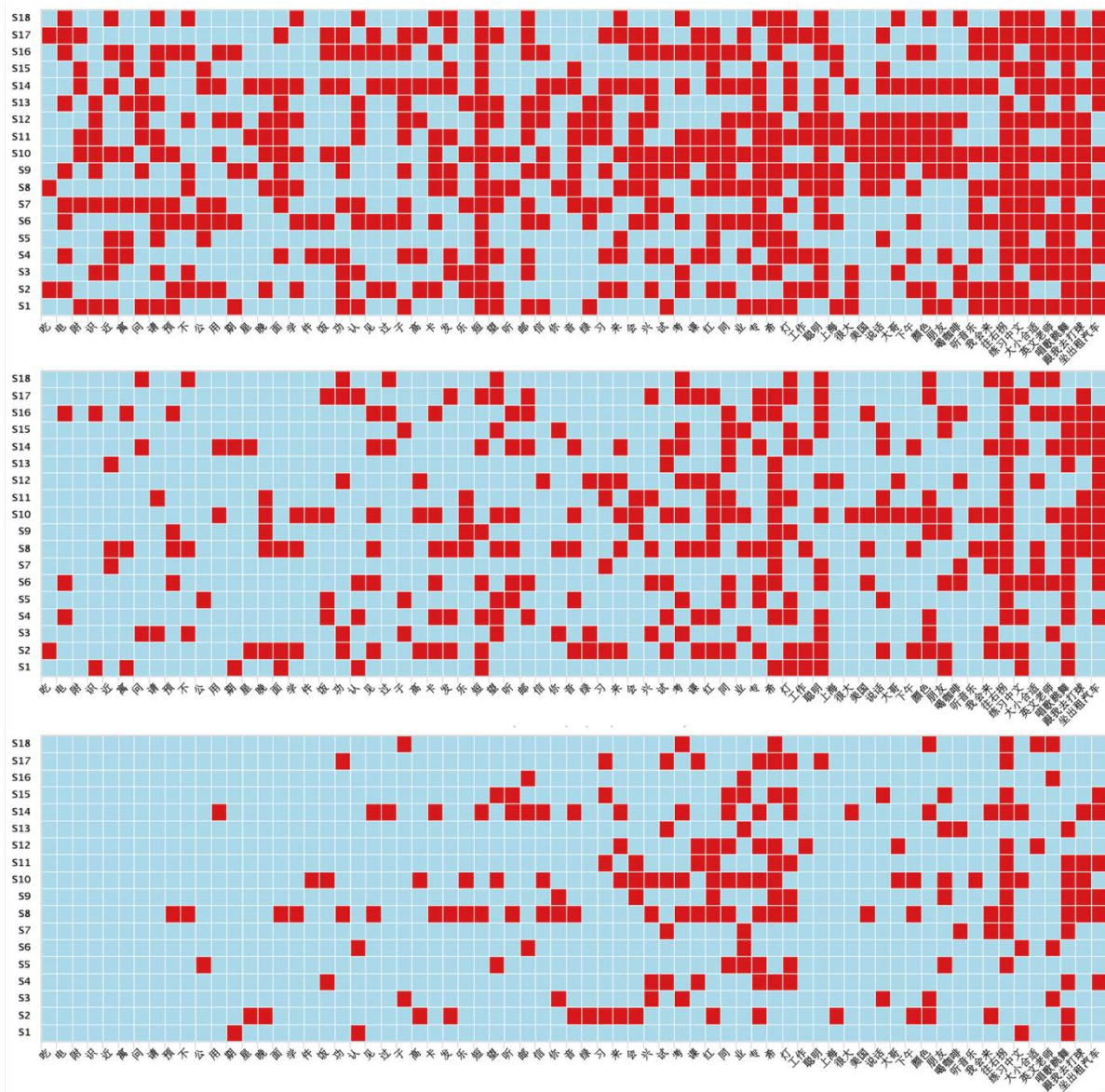
Figure 35. CSL learners' performance when they encountered each phrase at the first time (top), in the middle stage (middle), and at the last time (bottom) during the 3 weeks. Light blue color means that the certain participant achieved native-level proficiency on certain phrase, and red color means not. The phrases are sorted based on 1) the length; 2) the difficulty (i.e. determined by participants' performance when they met the phrases for the last time).

6.6.2 Performance Comparison with Native Speakers

In this session we investigated whether and to what extent CSL learners can achieve native speaker level proficiency on the given vocabulary set through their practice with ToneWars. We hypothesize that learners will be positively motivated if they can achieve native-level proficiency for a collection of phrases. The comparison is based on *PET* and the *"native level index"* defined earlier in the paper.

Figure 35 shows the visualizations of CSL learners' performance when they encountered each phrase at the first time (top), at the middle stage (middle, e.g., if one learner met a certain phrase for 6 times in total, we show the performance when she met the phrase at the 3rd i.e. (n+1)/2 time) and at the last time (bottom) during the 3 weeks. Light blue color means that the certain participant achieved native-level proficiency on certain phrase, and red color means not. The phrases are sorted based on 1) the length; 2) the difficulty (i.e. determined by the participants' performance when they met the phrases for the last time).

On average, the participants can achieve native level proficiency on 35.8 (51.9%) of phrases at the first time they met these phrases. In comparison, this number was increased to 58.2 (84.3%) at the last time they met these phrases during the 3-week study. The pairwise t-test shows that the difference is significant ($t(17)=-10.64$, $p<0.001$). We were also glad to find that there are 8 (11.6%) phrases that all participants can achieve native-level proficiency at the end of the study.

6.6.3 Visual Hints vs. Audio Hints

In this session we investigated the impacts of the two types of feedback (visual vs. audio) on learning gains. T-test showed a significant recall gain for tone and pinyin (post-test minus pre-test)

for both visual feedback and audio feedback conditions. For tone recall (Figure 36), the gain of visual feedback was 14.3 (22.5 vs. 36.8, t(17)=6.22, p<0.001), the gain of audio feedback is 15.5 (23.6 vs. 39.1, t(17)=10.87, p<0.001). Although audio hints led to 1.2 more tones learned than visual hints, the difference is not significant (14.3 vs. 15.5, t(17)=0.61, p=0.55). For pinyin recall (Figure 36), the gain of visual hints was 5.1 (33.2 vs. 38.3, t(17)=4.73, p<0.001), the gain of audio hints was 11.4 (32.3 vs. 43.7, t(17)=7.41, p<0.001). Audio hints led to 6.3 more pinyin learned than visual hints and the difference is significant (11.4 vs. 5.1, t(17)=4.26, p=0.001).

All participants reported that they preferred audio hints to visual hints for correcting and learning tones for new characters. Based on our observation and the interviews with participants, we found that the unique benefits of audio hints over visual hints are two-fold:

- *Listening to audio hints can improve CSL learner's ability to identify the correct tone by sound.*

Unlike visual hints with which users could easily see the exact gesture that must be drawn since the hint maps simply to the correct answer, audio hints required students to *"analyze the sound and determine the tone"* (S13) which involves complex mental processing. Even though it is easy for experienced learners who are familiar with the patterns of different tones, it is quite challenging for beginners. By analyzing the logs, we found that learners' such skill improved over time. Figure 37 shows the average number of audio hint played and the average number of guess in order to get the correct tone from an audio hint. In week 1, on average the CSL learners need to hear audio hint for 1.28 times and try 1.75 gestures in order to get the correct answer; in comparison, they need to hear the audio for 1.09 times and try 1.38 gestures in week 3. T-test shows that the difference is significant (1.28 vs. 1.09, t(17)=-1.81, p=0.048; 1.75 vs. 1.38, t(17)=1.95, p=0.013; respectively). This implies that their ability in recognizing tones from audio hints improved over time and the participants also noticed such improvement:

S12: *"Audio feedback gives a chance to the players to use their mind to figure out the right tone… I can do better later than at the beginning."*

S17: *"I enjoyed the audio feedback a lot more because it allowed me to hear a native speaker say the tone aloud. I thought this was beneficial because rather than it giving you the answer, it forces you to identify the tone by sound. Such skill is also very important."*

- *Audio hints can help CSL learners associate pronunciations and tones with characters, which can be treated as effective study tool.*

Audio hints not only tell the learners about the tones, but also the pronunciations of the characters. Such association can reinforce learner's aural knowledge of the characters and was appreciated by the learners:

S4: *"I liked audio feedback more because it allowed me to associate the correct pronunciation with the character."*

This may account for the significant difference in pinyin recall gains between visual hints and audio hints—audio hints also taught learners about the pronunciation of the characters, but visual hints did not. Learners also expressed their concerns when they were exposed to visual tone marks of characters that they did not know:

S3: *"no idea even though I got the tone mark if I don't know the character."*

S10: *"It doesn't make sense that only tell me about the tone without sound."*

Some learners used audio hint as learning materials and even read after that: users saw audio hints as closer to an effective, class-like study tool and thus took more advantage of these hints for their learning benefit.

S4: *"I did not know some of the words, so it taught me how to say them correctly."*

110

S11: *"You can hear how that character is supposed to sound like, and how that tone sounds, and I might be able to recognize it in conversation."*
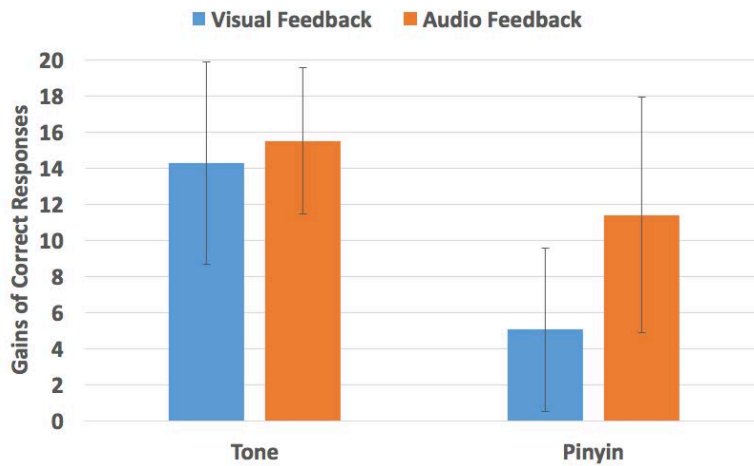
Figure 36. Tone/pinyin recall gains by feedback condition.

Even though all participants reported that they preferred audio hints for learning purpose, some of them mentioned that they preferred visual hints in gameplay because they were *"more straightforward and easier"* (S8), and they were *"equally helpful when you already know the pronunciation but not sure about the tone"* (S17).
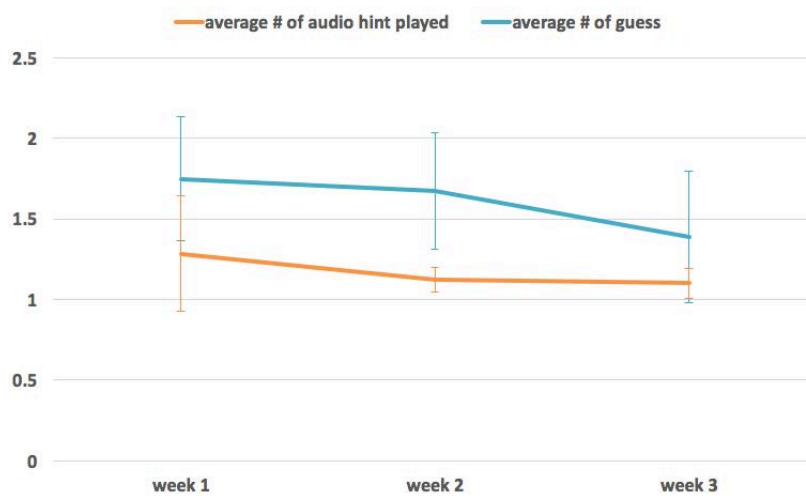
Figure 37. Average number of audio hint played and average number of guesses in order to recognize the correct tone.

6.6.4 Learners' Opinion towards Asynchronous Competition

In this session we investigated whether playing with a native speaker opponent rebuilt from user logs is still engaging for CSL learners. We first analyzed the user activity logs to retrieve all attacking behaviors (Figure 38) which could be the most direct indicators of how players engaged in such a competitive game. There was a significant increase on the average number of attacks that a user made (44.4 vs. 65.5, $t(17)=2.61$, $p=0.024$) the user made in a 5-minute session over the course of the 3-week study. Compared with 26.5 attacks per 5-minute in our previous lab study [76] in which CSL learners were competing with native speakers face-to-face, the numbers of attacks may suggest that CSL learners also engaged in the asynchronous gameplay. We attribute the increase of attacks to learners' better mastery of the game mechanics in the longitudinal study (3-hour game play vs. 40-minute game play in previous lab study [76]).

Learners' opinions towards asynchronous competition varies. 5 participants preferred synchronous competition—they explicitly mentioned that they would prefer playing against a real-life opponent in real time so that they can *"interact with native speaker directly"* (S4) and enjoy the *"real competition"* (S8). 4 participants preferred asynchronous game play because they *"would not feel so embarrassed when lose"* (S11), and they *"don't like compete too much"* (S3) due to personal playing style. The other 9 participants did not have preference—they reported that the motivation to play with the rebuilt native speaker opponent same as playing with a real-life opponent in real time. Some of them mentioned that they were more focusing on their *"learning rather than winning the game"* (S17).
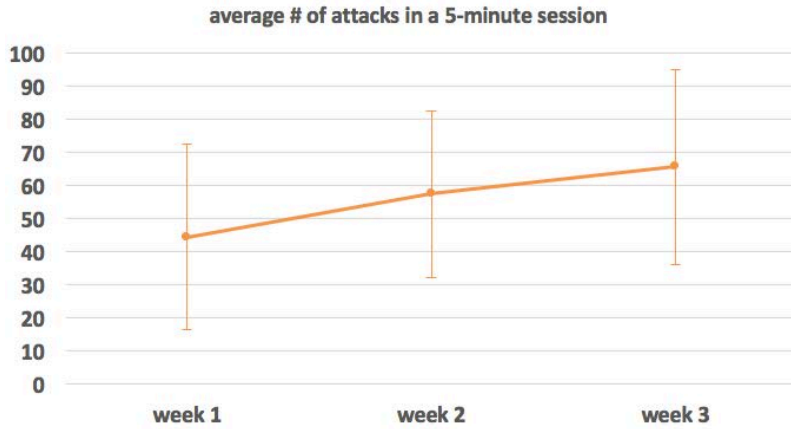
Figure 38. Average number of attacks in a play session.

Despite the variations in learners' opinions, all of them agreed that rebuilding the native speaker opponents from activity logs was the *"good enough solution"* (S8) to address the time zone problem. They also reported that it was engaging to play with the current native speaker opponent loaded in ToneWars which was "*pretty organic*" (S4).

6.6.5   Subjective Feedback
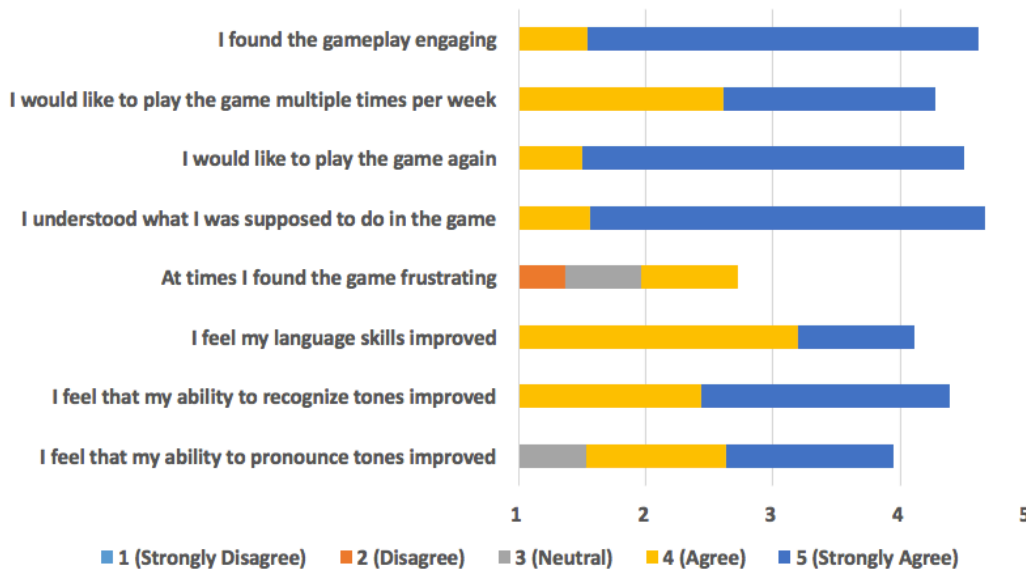
Figure 39. Subjective ratings on a 5-point Likert scale.

Overall, all participants reported positive and favorable experiences with ToneWars (Figure 39). Ratings were measured on a 5-Likert scale (1=*strongly disagree*, 5=*strongly agree*). Participants rated ToneWars' engagement as 4.61 (σ =0.61) and would like to play the game in the future (μ = 4.5, σ = 0.86).

At participants' requests, we installed ToneWars on 5 CSL learners' Android phones at the end of the study with customized vocabularies. Through the follow-up one month later, they reported that on average they spent 1.7 hours (min=0.5, max=3) per week on ToneWars gameplay, with high positiveness for the helpfulness of ToneWars.

6.6.6 Limitations

ToneWars is a mobile language learning game for informal learning. It is typical to not to set a control condition for longitudinal studies of informal language learning techniques in non-classroom environments (e.g. [54]). During the study, we decreased the potential for confounding by reducing the overlap between the learning materials loaded in ToneWars and the learning materials the participants had access in Chinese class. The learning materials in ToneWars were selected from the textbook "Integrated Chinese Level 1" (IC1). No participant was learning IC1 at the time of the study. While some of them might have occasional pronunciation feedback from Chinese class, none of them reported that they had additional tone training for the duration (dedicated tone training happens in the beginning two weeks, before learning the vocabularies. In the future, we will conduct in-the-wild longitudinal deployments to understand how well ToneWars works in real world scenarios.

6.7 DESIGN LESSONS AND FUTURE WORK

In this session we summarize the lessons we learned in the study and share these lessons with future designers and practitioners.

6.7.1 Native Speakers in Second Language Learning

In our paper, the terminology *"interaction"* means the competition and comparison of fine-grained language skills between CSL learners and native speakers in a mobile game, rather than "verbal" or "face-to-face" interactions. Talking directly to native speakers is beneficial, however, the direct communication can cause frustration and anxiety due to learners' perception of inequality when native speakers use some language that is beyond their comprehension levels, especially for novice learners. The role of native speakers in ToneWars is twofold: 1) a *benchmark* for fine-grained language mastery; 2) a *motivator* for language learning—learners are positively motivated towards further language learning when they know that they can achieve native-level proficiency for a subset of the language skills. We believe the use of native speakers as both a *benchmark* for language mastery and a *motivator* for language learning is a rich direction and can lead to interesting future research.

6.7.2 Learning via Multiple Modalities

All CSL learners reported that they preferred audio hints to visual hints for correcting and learning tones of new characters. Some of them attempted to recite characters following the audio hints. The major reason is that audio hints can allow them to *"associate the correct pronunciation with*

the character" (S4). Some CSL learners also suggest that ToneWars could *"show the pinyin together with tone mark"* (S3) in visual hints so that they can know the pronunciation of the character. S11 suggests that ToneWars can incorporate the *"English translation"* as well so that they can know the meaning of the character. We notice that the association between pinyin, tone, meaning, and character is very important for their character acquisition processes and each aspect will reinforce other aspects of their knowledge.

6.7.3 Character-Level vs. Phrase-Level Practice

Participants' preferences on character-level or phrase-level practice varied. Some learners (especially in novice level) thought character-level practice is easier and phrase-level practice is *"hard to memorize"* (S2). S12 mentioned that *"phrase is more frustrating if knew nothing in the phrase"*. At the same time, some of them expressed that they would *"like to learn in phrases"* (S15) since it can provide them more context information regarding how to use the characters in phrases. In comparison, S1, with more than 3-year Mandarin learning experience and 6-month living experience in China, found that sometimes it was challenging to recognize single characters than phrases. As an example, she mentioned that she can correctly recognize and read "希望" (hope) as a phrase, but cannot recognize the single characters if they do not appear together because she lost the context. She expressed that she would like to take more *"practice with single characters"* after she noticed this problem.

6.7.4 Fine-Grained Feedback on Language Mastery

Through fine-grained modeling of native speakers' language skills, CSL learners can compete with them on phrase level tone recall tasks. Currently, they can see the real-time scores and the preview of their opponents' stacks to infer their relative performance and success. In the future, we plan to incorporate more explicit and fine-grained feedback on language mastery—once a phrase is eliminated and the PET reaches the NLI, ToneWars provides a bonus and conveys the player about their success. We believe that such fine-grained modeling and feedback on language mastery could enhance language learners' self-confidence and highly motivate them towards further learning. We also believe that this approach and the corresponding insights are generalizable to other language learning applications beyond Chinese.

6.8   SUMMARY

We present principled research to make ToneWars *scalable* and *sustainable*. First, we propose a scalable approach to enable asynchronous competition and skill comparison by modeling both the interaction patterns and language skills of native speakers. We conduct fine-grained modeling of native speakers' language skills (e.g. phrase level tone recall) and use it as the goal for learners to achieve "bite-sized" native level mastery. Second, we propose a novel metric to quantify whether a CSL learner achieves native level proficiency for a specific tone at a given moment, and use such fine-grained language mastery as a sustainable motivating factor for further learning. In a longitudinal study with 18 CSL learners, we found that such asynchronous modeling can motivate learners in a sustained manner. We also observed significantly improved learning gains (e.g.,

average gains of 29.7 tones and 16.4 syllables). Participants achieved native level proficiency on

58.2 out of 69 phrases at the end of the study.

7.0   CONCLUSIONS

7.1   SUMMARY OF CONTRIBUTIONS

This dissertation explores the usage of intelligent user interfaces to facilitate the efficient and effective adoption of the tried-and-true pedagogies *at scale*. We present holistic approaches that are inspired by proven pedagogies, can address practical challenges, and are tested in real-world scenarios. Specifically, we make the following contributions:

Chapter 3 presents **MindMiner** [56, 57]: an interactive data exploration and visualization system for instructors to understand student *peer review* data and generate customized feedback in a scalable manner. MindMiner collects and quantifies instructors' subjective knowledge on entity similarity via *mixed-initiative interfaces* and novel *machine learning algorithms*. MindMiner then uses such knowledge in clustering tasks to improve data exploration efficiency. In a 12-subject user study, we found that MindMiner can capture the implicit similarity measurement from users and can improve users' understanding of students' performance. Our contributions include:

- We propose two interaction techniques, *active polling with uncertainty* and *example-based constraints collection*, to collect, visualize, and manage implicit, subjective domain knowledge by scaffolding end-users incrementally.

- We introduce an improved distance metric learning algorithm that takes into account input ambiguity and avoids trivial solutions1 in existing algorithms.

- We present effective *active learning* heuristics and corresponding interface design to collect pairwise constraints at both entity and group levels. We show in a 12-subject controlled study that our design can significantly enhance the clustering relevance.

- We present an interactive data exploration and visualization system, MindMiner, to help end-users externalize domain knowledge and improve data exploration efficiency via distance metric learning. To our knowledge, this is the first interactive system that provides both algorithm and interface level support for handling inconsistent, ambiguous domain knowledge via distance metric learning.

Chapter 4 presents **BayesHeart** [55]: a commodity-camera-based photoplethysmography (PPG) sensing and probabilistic-based heart rate monitoring algorithm on unmodified smartphones. When integrated with MOOC mobile client applications, BayesHeart can capture and collect learners' heart rates implicitly when they watch lecture videos. Such information is the foundation of learner attention/affect modeling, which enables a *'sensorless'* and *scalable* feedback channel from students to instructors. Our contributions include:

- We present BayesHeart, a probabilistic algorithm that extracts both heart rates and distinct phases of the cardiac cycle directly from noisy, intermittent ROI signals captured by camera phones. We released the source code of BayesHeart under BSD license at http://mips.lrdc.pitt.edu/bayesheart

- By decoupling existing camera based heart rate monitoring techniques into two steps, i.e. noisy reduction and cardiac pulse counting, we identified the design space and compared existing technologies side-by-side highlighting both their relationships and new opportunities.

- In a 20-subject experiment, we systematically evaluated the state-of-the-art algorithms covering the design space regarding accuracy and latency performance.

120

Chapter 5 presents **CourseMIRROR** [58, 59, 109]: a mobile learning system that uses natural language processing (NLP) techniques to enhance large classroom instructor-student interactions via streamlined and scaffolded *reflection prompts*. CourseMIRROR can 1) automatically remind and collect students' in-situ written reflections after each lecture; 2) continuously monitor the quality of a student's reflection at composition time and generate helpful feedback to scaffold reflection writing; 3) summarize the reflections and present the most significant ones to both instructors and students. CourseMIRROR is freely available for classroom usage at: http://www.coursemirror.com. Our contributions include:

- We present CourseMIRROR, a scalable mobile learning system that uses NLP techniques to facilitate the collection and use of high quality responses to *reflection prompts* in large classrooms.

- We show that the interactive reflection quality feedback feature can scaffold students to write concrete and specific reflections. Our algorithms are scalable to courses in diverse topics and robust to cold start.

- We share our insights and lessons learned from eight semester-long deployments.

Chapter 6 presents **ToneWars** [60]: an educational game connecting Chinese as a Second Language (CSL) learners with native speakers via mobile gameplay. CSL Learners can practice tone recall, perception and production by competing with native speakers in ToneWars. We propose a scalable approach to enable authentic competition and skill comparison with native speakers by modeling both the interaction patterns and language skills of native speakers asynchronously. Our contributions include:

- We demonstrate the motivational power and feasibility of the fine-grained modeling of native-speaker skills (e.g. phrase level tone recall) and uses it as the goal for learners to achieve "bite-

121

sized" native-speaker level mastery. We find that this approach can motivate learners in a sustainable manner.

- We propose a scalable approach to enable authentic competition and skill comparison with native speakers by modeling both the interaction patterns and language skills of native speakers asynchronously.

- We prove the effectiveness of such modeling in a longitudinal setting.

7.2   LIMITATIONS AND FUTURE WORK

Assess learning outcomes through large scale deployments. Improved learning outcome is the holy grail of educational systems. However, assessing the impact of software interventions on learning outcomes *in real-world settings* is always challenging. We have evaluated the learning gains of using ToneWars in a longitudinal study. However, the study was still conducted in lab settings. We have deployed MindMiner and CourseMIRROR in real-world classrooms, but have not completed a controlled study to formally evaluate the learning outcomes. Challenges include: 1) these systems were involving through iterative design processes rather than being created to be final products—the dynamic nature made it hard to freeze all the features and run a controlled deployment during the process; 2) we need courses with high opt-in rate (ideally 40 or more students per condition) and multiple parallel sessions to get the statistical power needed to analyze the learning outcome in semester-long deployments. In the future, we plan to conduct large-scale class deployments with control groups to evaluate the learning outcomes of the systems included in this dissertation.

Design for better motivation. Designing the right tool cannot guarantee the effective and massive use of this tool—motivating usage (for both students and instructors) is essential, especially for educational systems. For most learning systems, the benefits come from continuous and sustained usage, while being not immediately apparent. While we built the systems presented in this dissertation, we designed features to motivate both instructors and students, e.g., the engaging game design in ToneWars, the lecture-time-triggered push notification and the quality feedback feature in CourseMIRROR. In addition, we also experimented with different external incentives and demonstrated their effectiveness to encourage usage, e.g., monetary incentives (ToneWars, CourseMIRROR) and course incentives (CourseMIRROR). In the future, we plan to further explore how to design for better motivation. Specifically, 1) how can we make instructors and students always perceive the benefits, even at the early stage of usage? 2) can we provide increased values with the current systems and infrastructures so that they can get more benefits in the adoption, e.g., by enabling the "clicker" function on CourseMIRROR?

Scaffold instructors to go beyond scalable assessment. Currently our approaches mainly focus on enabling instructors to have scalable *assessment* of their students, e.g., via peer review understanding (MindMiner), implicit physiological signal sensing (BayesHeart), and summaries of student reflections (CourseMIRROR). We believe that there still exist opportunities to scaffold and facilitate instructors to go one step further—converting the scalable *assessment* to concrete *actions* and *interventions* in the follow-up teaching activities. For example, how to facilitate instructors to generate personalized feedback after they get the relevant clustering results via MindMiner? How to facilitate instructors to address students' difficulties and misconceptions after reading the summaries generated by CourseMIRROR? We plan to explore techniques to scaffold

123

and facilitate instructors to make effective interventions based on their understanding of the performance and needs of their students.

BIBLIOGRAPHY

1. Khalil Al-Mekhlafi, Xiangpei Hu, and Ziguang Zheng. "An approach to context-aware mobile Chinese language learning for foreign students." Mobile Business, 2009. ICMB 2009. Eighth International Conference on. IEEE, 2009.

2. Vincent Aleven. "Helping students to become better help seekers: Towards supporting metacognition in a cognitive tutor." German-USA Early Career Research Exchange Program: Research on Learning Technologies and Technology-Supported Education, Tubingen, Germany (2001).

3. Vincent Aleven, and Kenneth R. Koedinger. "An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor." Cognitive Science 26, no. 2 (2002): 147-179.

4. Vincent Aleven, Octav Popescu, and Kenneth R. Koedinger. "Towards tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor." In Proceedings of Artificial Intelligence in Education, pp. 246-255. 2001.

5. John R. Anderson, Albert T. Corbett, Kenneth R. Koedinger, and Ray Pelletier. "Cognitive tutors: Lessons learned." The journal of the learning sciences 4.2 (1995): 167-207.

6. Rie Kubota Ando, and Tong Zhang. "A high-performance semi-supervised learning method for text chunking." In Proceedings of the 43rd annual meeting on association for computational linguistics, pp. 1-9. Association for Computational Linguistics, 2005.

7. Ivon Arroyo, Beverly Park Woolf, Winslow Burelson, Kasia Muldner, Dovan Rai, and Minghui Tai. "A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect." International Journal of Artificial Intelligence in Education, 24(4), 387-426.

8. Susan Aud, Thomas Nachazel, Sidney Wilkinson-Flicker, and Allison Dziuba. "The condition of education 2013." Government Printing Office, 2013.

9. John R. Baird, Peter J. Fensham, Richard F. Gunstone, and Richard T. White. "The importance of reflection in improving science teaching and learning." Journal of research in Science Teaching 28, no. 2 (1991): 163-182.

10. Guha Balakrishnan, Fredo Durand, and John Guttag. "Detecting pulse from head motions in video." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013.

11. K. Banitsas, P. Pelegris, T. Orbach, D. Cavouras, K. Sidiropoulos, and S. Kostopoulos. "A simple algorithm to monitor hr for real time treatment applications." 2009 9th International Conference on Information Technology and Applications in Biomedicine. IEEE, 2009.

12. Jennifer S. Beaudin, Stephen S. Intille, Emmanuel Munguia Tapia, Randy Rockinson, and Margaret E. Morris. "Context-sensitive microlearning of foreign language vocabulary on a mobile device." Ambient Intelligence. Springer Berlin Heidelberg, 2007. 55-72.

13. Joshua E. Blumenstock. "Size matters: word count as a measure of quality on wikipedia." In Proceeding of the 17th international conference on World Wide Web - WWW '08, ACM Press (2008), 1095–1096.

14. Kees Bot. "The psycholinguistics of the Output Hypothesis." Language Learning, 46, 529-555, 1996.

15. David Boud, Rosemary Keogh, and David Walker. "Promoting reflection in learning: A Model." Boundaries of adult learning 1 (2013): 32.

16. David Boud, Rosemary Keogh, and David Walker. Reflection: Turning experience into learning. Routledge 2013.

17. Michael Brooks, Sumit Basu, Charles Jacobs, and Lucy Vanderwende. "Divide and correct: Using clusters to grade short answers at scale." Proceedings of the first ACM conference on Learning@ scale conference. ACM, 2014.

18. Marika de Bruijne, and Arnaud Wijnant. "Comparing survey results obtained via mobile devices and computers: an experiment with a mobile web survey on a heterogeneous group of mobile devices versus a computer-assisted web survey." Social Science Computer Review (2013): 0894439313483976.

19. Ray A. Burnstein, and Leon M. Lederman. "Using wireless keypads in lecture classes." The Physics Teacher 39, no. 1 (2001): 8-11.

20. T. D. Buskirk, and Charles Andrus. "Online surveys aren't just for computers anymore! Exploring potential mode effects between smartphone vs. computer-based online surveys." AAPOR Annual Conference. 2012.

21. Deborah L. Butler, and Philip H. Winne. "Feedback and self-regulated learning: A theoretical synthesis." Review of educational research 65.3 (1995): 245-281.

22. J. E. Caldwell. "Clickers in the large classroom: Current research and best-practice tips." CBE-Life Sciences Education 6.1 (2007): 9-20.

23. J. Caldwell, J. Zelkowski, and M. Butler. "Using personal response systems in the classroom." In WVU Technology Symposium, April, vol. 11, p. 2006. 2006.

24. Rafael A. Calvo, and Sidney D'Mello. "Affect detection: An interdisciplinary review of models, methods, and their applications." Affective Computing, IEEE Transactions on 1.1 (2010): 18-37.

25. Julia Cambre, Chinmay Kulkarni, Michael S. Bernstein, and Scott R. Klemmer. "Talkabout: small-group discussions in massive global classes." Proceedings of the first ACM conference on Learning@ scale conference. ACM, 2014.

26. Nan Cao, David Gotz, Jimeng Sun, and Huamin Qu. "DICON: Interactive Visual Analysis of Multidimensional Clusters." In IEEE Transactions on Visualization and Computer Graphics, 17(12), pp. 2581-2590. IEEE Press, New York (2011).

27. Cardiograph for iOS, https://itunes.apple.com/us/app/cardiograph/id441079429?ls=1&mt=8

28. Scott Carter, Jennifer Mankoff, and Jeffrey Heer. "Momento: support for situated ubicomp experimentation." In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 125-134. ACM, 2007.

29. Nadire Cavus, and Dogan Ibrahim. "m-Learning: An experiment in using SMS to support learning new English language words." British journal of educational technology, 40.1 (2009): 78-91.

30. Michelene TH Chi. "Active-constructive-interactive: A conceptual framework for differentiating learning activities." Topics in Cognitive Science 1.1 (2009): 73-105.

31. Michelene TH Chi, Nicholas Leeuw, Mei‑Hung Chiu, and Christian LaVancher. "Eliciting self-explanations improves understanding." Cognitive science 18.3 (1994): 439-477.

32. Kwangsu Cho, and Christian D. Schunn. "Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system." Computers & Education 48.3 (2007): 409-426.

33. Jongyoon Choi, and Ricardo Gutierrez-Osuna. "Using heart rate monitors to detect mental stress." In BSN 2009. Sixth International Workshop, 219-223.

34. Douglas A. Coast, Richard M. Stern, Gerald G. Cano, and Stanley A. Briller. "An approach to cardiac arrhythmia analysis using hidden Markov models." Biomedical Engineering, IEEE Transactions on 37.9 (1990): 826-836.

35. Derrick Coetzee, Armando Fox, Marti A. Hearst, and Bjoern Hartmann. "Chatrooms in MOOCs: all talk and no action." Proceedings of the first ACM conference on Learning@ scale conference. ACM, 2014.

36. David Cohn, Les Atlas, and Richard Ladner. "Improving generalization with active learning." Machine Learning. 15(2), pp. 201-221, 1994.

37. David Cohn, Rich Caruana, and Andrew McCallum. "Semi-supervised clustering with user feedback." Constrained Clustering: Advances in Algorithms, Theory, and Applications, 4(1), 17-32.

38. Linda J. Collins "Livening up the classroom: Using audience response systems to promote active learning." Medical reference services quarterly 26.1 (2007): 81-88.

39. Cristina Conati, and Kurt Vanlehn. "Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation." International Journal of Artificial Intelligence in Education 11 (2000): 389-415.

40. Stéphane Cook, Mario Togni, Marcus C. Schaub, Peter Wenaweser, and Otto M. Hess. "High heart rate: a cardiovascular risk factor?" European heart journal, vol. 27(20), 2006.

41. Kenneth H. Cooper, Michael L. Pollock, Randolph P. Martin, Steve R. White, Ardell C. Linnerud, and Andrew Jackson. "Physical Fitness Levels vs Selected Coronary Risk Factors A Cross-Sectional Study" The Journal of the American Medical Association (JAMA), vol. 236, No. 2, July, 1976.

42. Catherine H. Crouch, and Eric Mazur. "Peer instruction: Ten years of experience and results." American journal of physics 69.9 (2001): 970-977.

43. Sidney D'Mello, Rosalind W. Picard, and Arthur Graesser. "Toward an affect-sensitive AutoTutor." IEEE Intelligent Systems 4 (2007): 53-61.

44. Eden Dahlstrom, J. D. Walker, and Charles Dziuban "ECAR study of undergraduate students and information technology." 2015. Educause Center for Applied Research.

45. Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. "Co-clustering based classification for out-of-domain documents." In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 210-219. ACM, 2007.

46. V. S. Damodharan, and V. Rengarajan, "Innovative methods of teaching." Learning Technologies and Mathematics Middle East Conference, Sultan Qaboos University, Muscat, Oman. 2007.

47. David Dearman, and Khai Truong. "Evaluating the implicit acquisition of second language vocabulary using a live wallpaper." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2012.

48. Marie Desjardins, James MacGlashan, and Julia Ferraioli. "Interactive visual clustering." In Proceedings of the 12th international conference on Intelligent user interfaces, pp. 361-364. ACM, New York (2007).

49. Susan T. Dumais. "Latent semantic analysis." Annual review of information science and technology 38, no. 1 (2004): 188-230.

50. Nathan Dummitt. Chinese Through Tone & Color. Hippocrene Books, 2008.

51. Jennifer G. Dy, and Carla E. Brodley. "Visualization and interactive feature selection for unsupervised data." In Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 360-364. ACM, New York (2000).

52. Darren Edge, Stephen Fitchett, Michael Whitney, and James Landay. "MemReflex: adaptive flashcards for mobile microlearning." In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services, pp. 431-440. ACM, 2012.

53. Darren Edge, Elly Searle, Kevin Chiu, Jing Zhao, and James A. Landay "MicroMandarin: mobile language learning in context." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 3169-3178. ACM, 2011.

54. Darren Edge, Kai-Yin Cheng, Michael Whitney, Yao Qian, Zhijie Yan, and Frank Soong. "Tip tap tones: mobile microtraining of mandarin sounds." In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services, pp. 427-430. ACM, 2012.

55. Xiangmin Fan, and Jingtao Wang. "BayesHeart: A Probabilistic Approach for Robust, Low-Latency Heart Rate Monitoring on Camera Phones." Proceedings of the 20th International Conference on Intelligent User Interfaces. ACM, 2015.

56. Xiangmin Fan, Youming Liu, Nan Cao, Jason Hong, and Jingtao Wang. "MindMiner: A Mixed-Initiative Interface for Interactive Distance Metric Learning." Human-Computer Interaction–INTERACT 2015. Springer International Publishing, 2015. 611-628.

57. Xiangmin Fan, Youming Liu, Nan Cao, Jason Hong, and Jingtao Wang. "MindMiner: Quantifying Entity Similarity via Interactive Distance Metric Learning." Proceedings of the 20th International Conference on Intelligent User Interfaces Companion. ACM, 2015.

58. Xiangmin Fan, Wencan Luo, Muhsin Menekse, Diane Litman, and Jingtao Wang. "CourseMIRROR: Enhancing large classroom instructor-student interactions via mobile interfaces and natural language processing." In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems. ACM, 2015.

59. Xiangmin Fan, Wencan Luo, Muhsin Menekse, Diane Litman, and Jingtao Wang. "Scaling reflection prompts in large classrooms via mobile interfaces and natural language processing." Proceedings of the 22nd International Conference on Intelligent User Interfaces. ACM, 2017.

60. Xiangmin Fan, Wencan Luo, and Jingtao Wang. "Mastery Learning of Second Language Through Asynchronous Modeling of Native Speakers in a Collaborative Mobile Game." In Proceedings of the 35th Annual ACM Conference on Human Factors in Computing Systems. ACM, 2017.

61. Kyle Feldscher (Sep 25th, 2015). "Obama wants 1 million Americans learning Chinese by 2020". http://www.washingtonexaminer.com/article/2572865

62. Kim Fox, Jeffrey S. Borer, A. John Camm, Nicolas Danchin, Roberto Ferrari, Jose L. Lopez Sendon, Philippe Gabriel Steg. "Resting heart rate in cardiovascular disease". Journal of the American College of Cardiology, 50(9), 823-830.

63. Jon Froehlich, Mike Y. Chen, Sunny Consolvo, Beverly Harrison, and James A. Landay. "MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones." In Proceedings of the 5th international conference on Mobile systems, applications and services, pp. 57-70. ACM, 2007.

64. I-Ping P. Fu. "Student approaches to learning Chinese vocabulary." doctoral dissertation, Virginia Polytechnic Institute and State University (2005).

65. Marc Garbey, Nanfei Sun, Arcangelo Merla, and Ioannis Pavlidis. "Contact-free measurement of cardiac pulse based on the analysis of thermal imagery." Biomedical Engineering, IEEE Transactions on 54.8 (2007): 1418-1426.

66. Elena L. Glassman, Juho Kim, Andrés Monroy-Hernández, and Meredith Ringel Morris. "Mudslide: A Spatially Anchored Census of Student Confusion for Online Lecture Videos." Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM, 2015.

67. Robert Godwin-Jones. "Emerging technologies from memory palaces to spacing algorithms: approaches to second language vocabulary learning." Language, Learning & Technology 14.2 (2010): 4-11.

68. GoSoapBox: a web-based clicker tool. http://www.gosoapbox.com

69. Grace Kena, Lauren Musu-Gillette, Jennifer Robinson, Xiaolei Wang, Amy Rathbun, Jijun Zhang, Sidney Wilkinson-Flicker, Amy Barmer, and Erin Dunlop Velez Velez. "The Condition of Education 2015". National Center for Education Statistics (2015).

70. Mathew J. Gregoski, Martina Mueller, Alexey Vertegel, Aleksey Shaporev, Brenda B. Jackson, Ronja M. Frenzel, Sara M. Sprehn, and Frank A. Treiber. "Development and validation of a smartphone heart rate acquisition application for health promotion and wellness telehealth applications." International journal of telemedicine and applications,2012, 1.

71. Domenico Grimaldi, Yuriy Kurylyak, Francesco Lamonaca, and Alfonso Nastro. "Photoplethysmography detection by smartphone's videocamera." In Proc. IDAACS 2011, 488-491.

72. Philip J. Guo, Juho Kim, and Rob Rubin. "How video production affects student engagement: An empirical study of mooc videos." Proceedings of the first ACM conference on Learning@ scale conference. ACM, 2014.

73. Teng Han, Xiang Xiao, Lanfei Shi, John Canny, and Jingtao Wang. "Balancing accuracy and fun: designing camera based mobile games for implicit heart rate monitoring." In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 847-856. ACM, 2015.

74. Noriko Hara, Curtis Jay Bonk, and Charoula Angeli. "Content analysis of online discussion in an applied educational psychology course." Instructional science 28.2 (2000): 115-152.

75. William S. Harwood. "The one-minute paper." Journal of Chemical Education 73.3 (1996): 229.

76. Andrew Head, Yi Xu, and Jingtao Wang. "Tonewars: Connecting language learners and native speakers through collaborative mobile games." In International Conference on Intelligent Tutoring Systems, pp. 368-377. Springer International Publishing, 2014.

77. Yifen Huang, and Tom M. Mitchell. "Exploring Hierarchical User Feedback in Email Clustering." In EMAIL'08: Proceedings of the Workshop on Enhanced Messaging-AAAI, pp. 36-41. AAAI, Menlo Park (2008).

78. Nicholas P. Hughes, Lionel Tarassenko, and Stephen J. Roberts. "Markov Models for Automated ECG Interval Analysis." In NIPS. 2003.

79. Instant Heart Rate for iOS, https://itunes.apple.com/app/instant-heart-rate-measure/id395042892?mt=8

80. InstFeedback. http://www.instfeedback.com

81. Kathleen F. Janz, JEFFREY D. Dawson, and Larry T. Mahoney. "Tracking physical fitness and physical activity from childhood to adolescence: the Muscatine study." Medicine and Science in Sports and Exercise, vol. 32(7), 2000.

82. Magnus Thorsten Jensen, Poul Suadicani, Hans Ole Hein, and Finn Gyntelberg. "Elevated resting heart rate, physical fitness and all-cause mortality: a 16-year follow-up in the Copenhagen Male Study." Heart, 99(12), 882-887.

83. Nicholas Johnson, Phil Oliff, and Erica Williams. "An update on state budget cuts." Center on Budget and Policy Priorities. Updated February 9 (2011).

84. W. Lewis Johnson, and Jeff Rickel. "Steve: An animated pedagogical agent for procedural training in virtual environments." ACM SIGART Bulletin 8.1-4 (1997): 16-21.

85. E. Jonathan, and Martin Leahy. "Investigating a smartphone imaging unit for photoplethysmography." Physiological measurement, 31(11), N79.

86. Katy Jordan. "Initial trends in enrolment and completion of massive open online courses." The International Review Of Research In Open And Distributed Learning 15.1 (2014).

87. Sally Jordan, and Tom Mitchell. "e-Assessment for learning? The potential of short-answer free-text questions with tailored feedback." British Journal of Educational Technology 40.2 (2009): 371-385.

88. Matthew Kam, Divya Ramachandran, Varun Devanathan, Anuj Tewari, and John Canny. "Localized iterative design for language learning in underdeveloped regions: the PACE

framework." In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 1097-1106. ACM, 2007.

89. Sandra Katz, Alan Lesgold, Edward Hughes, Daniel Peters, Gary Eggan, Maria Gordin, and Linda Greenberg. "Sherlock 2: An intelligent tutoring system built on the lrdc framework." Facilitating the development and use of interactive learning environments. ERLBAUM (1998).

90. Ayako Kawase. "Second language acquisition and synchronous computer mediated communication." Tesol & Applied Linguistics 6.2 (2006): 1-27.

91. Karen Kear. "Peer learning using asynchronous discussion systems in distance education." Open Learning: The Journal of Open, Distance and e-Learning, 19(2), 151-164.

92. Gregor E. Kennedy, and Quintin I. Cutts. "The association between students' use of an electronic voting system and their learning outcomes." Journal of Computer Assisted Learning 21.4 (2005): 260-268.

93. Juho Kim, Philip J. Guo, Daniel T. Seaton, Piotr Mitros, Krzysztof Z. Gajos, and Robert C. Miller. "Understanding in-video dropouts and interaction peaks in online lecture videos." Proceedings of the first ACM conference on Learning@ scale conference. ACM, 2014.

94. Juho Kim, Philip J. Guo, Carrie J. Cai, Shang-Wen Daniel Li, Krzysztof Z. Gajos, and Robert C. Miller. "Data-driven interaction techniques for improving navigation of educational videos." Proceedings of the 27th annual ACM symposium on User interface software and technology. ACM, 2014.

95. Juho Kim, Elena L. Glassman, Andrés Monroy-Hernández, and Meredith Ringel Morris. "RIMES: Embedding interactive multimedia exercises in lecture videos." In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 1535-1544. ACM, 2015.

96. Constantine Kiriloff. "On the auditory perception of tones in Mandarin." Phonetica 20.2-4 (1969): 63-67.

97. Kenneth R. Koedinger, John R. Anderson, William H. Hadley, and Mary A. Mark. "Intelligent tutoring goes to school in the big city." (1997).

98. Stephen D Krashen. "The input hypothesis: Issues and implications." Addison-Wesley Longman Ltd, 1985.

99. Chinmay E. Kulkarni, Michael S. Bernstein, and Scott R. Klemmer. "PeerStudio: Rapid Peer Feedback Emphasizes Revision and Improves Performance." Proceedings from The Second (2015) ACM Conference on Learning@ Scale. 2015.

100. Anuj Kumar, Pooja Reddy, Anuj Tewari, Rajat Agrawal, and Matthew Kam. "Improving literacy in developing countries using speech recognition-supported games on mobile devices." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1149-1158. ACM, 2012.

101. Reed Larson, and Mihaly Csikszentmihalyi. "The experience sampling method." New Directions for Methodology of Social & Behavioral Science, 1983.

102. R. Dwight Laws, Scott L. Howell, and Nathan K. Lindsay. "Scalability in Distance Education: Can We Have Our Cake and Eat it Too?" Online Journal of Distance Learning Administration 6.4 (2003).

103. Jonathan Leather. "F0 Pattern Inference in the Perceptual Acquisition of Second Language Tone in Sound Patterns in Second Language Acquisition." Studies on Language Acquisition (SOLA) 5 (1987): 59-80.

104. Jonathan Leather. "Perceptual and productive learning of Chinese lexical tone by Dutch and English speakers." New sounds 90 (1990): 72-97.

105. Lina Lee. "Learners' perspectives on networked collaborative interaction with native speakers of Spanish in the US." Language Learning & Technology 8, no. 1 (2004): 83-100.

106. Lili Liu. (Jun 27th 2011). "Chinese language proficiency test becoming popular in Mexico." http://news.xinhuanet.com/english2010/china/2011-06/27/c_13951048.htm

107. Tania Lombrozo. "The structure and function of explanations." Trends in cognitive sciences 10.10 (2006): 464-470.

108. Michael H. Long. "Native speaker/non-native speaker conversation in the second language classroom." University of Hawai'i Working Papers in English as a Second Language 2 (1) (1983).

109. Wencan Luo, Xiangmin Fan, Muhsin Menekse, Jingtao Wang, and Diane J. Litman. "Enhancing instructor-student and student-student interactions with mobile interfaces and summarization." Proceedings of NAACL-HLT. 2015.

110. Wencan Luo, and Diane Litman. "Determining the Quality of a Student Reflective Response." The Twenty-Ninth International FLAIRS Conference. 2016.

111. Wencan Luo, and Diane Litman. "Summarizing Student Responses to Reflection Prompts." In Proceedings of EMNLP, 2015.

112. Gloria Mark, Yiran Wang, and Melissa Niiya. "Stress and Multitasking in Everyday College Life: An Empirical Study of Online Activity." In Proc. CHI 2014.

113. Ellen M. Markman. "Realizing that you don't understand: Elementary school children's awareness of inconsistencies." Child development (1979): 643-655.

114. Mark Maybury. "Intelligent user interfaces: an introduction." In Proceedings of the 4th international conference on Intelligent user interfaces, pp. 3-4. ACM, 1998.

115. Beate H. McGhee, and Elizabeth J. Bridges. "Monitoring arterial blood pressure: what you may not know." Critical Care Nurse, 22(2), 60-79.

116.   Noel McIntosh. "Why do we lecture?" (1996).

117.   Muhsin Menekse, Glenda Stump, Stephen J. Krause, and Michelene TH Chi. "The effectiveness of students' daily reflections on learning in engineering context." In 118th ASEE Annual Conference and Exposition. 2011.

118.   Chet Meyers, and Thomas B. Jones. "Promoting Active Learning. Strategies for the College Classroom." Jossey-Bass Inc., Publishers, 350 Sansome Street, San Francisco, CA 94104, 1993.

119.   Joan Middendorf, and Alan Kalish. "The "change-up" in lectures." Natl. Teach. Learn. Forum. Vol. 5. No. 2. 1996.

120.   Michael Mitchell, and Michael Leachman. "Years of cuts threaten to put college out of reach for more students." Center on Budget and Policy Priorities (2015): 1-26.

121.   Michael Mohler, Razvan Bunescu, and Rada Mihalcea. "Learning to grade short answer questions using semantic similarity measures and dependency graph alignments." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011.

122.   Laura M. Morett, and Li-Yun Chang. "Emphasizing sound and meaning: pitch gestures enhance Mandarin lexical tone acquisition." Language, Cognition, and Neuroscience, 30:3, 347-353.

123.   Frederick Mosteller. "The 'Muddiest Point in the Lecture'as a feedback device." On Teaching and Learning: The Journal of the Harvard-Danforth Center 3 (1989): 10-21.

124.   Catherine Mulryan-Kyne. "Teaching large classes at college and university level: Challenges and opportunities." Teaching in Higher Education 15.2 (2010): 175-185.

125.   Huy Nguyen, Wenting Xiong, and Diane Litman. "Instant Feedback for Increasing the Presence of Solutions in Peer Reviews." In Proceedings Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations (NAACL-HLT), pp. 6-10

126.   Scott E. Page. "Model Thinking." https://www.coursera.org/learn/model-thinking

127.   Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. "Cross-domain sentiment classification via spectral feature alignment." In Proceedings of the 19th international conference on World wide web, pp. 751-760. ACM, 2010.

128.   Panagiotis Pelegris, K. Banitsas, T. Orbach, and Kostas Marias. "A novel method to detect heart beat rate using a mobile phone." In Proc. EMBC 2010, 5488-5491.

129.   Adam Perer, and Ben Shneiderman. "Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis." In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp. 265-274. ACM, New York (2008)

130. Phuong Pham, and Jingtao Wang. "Adaptive review for mobile MOOC learning via implicit physiological signal sensing." Proceedings of the 18th ACM International Conference on Multimodal Interaction. ACM, 2016.

131. Phuong Pham, and Jingtao Wang. "AttentiveLearner: Improving Mobile MOOC Learning via Implicit Heart Rate Tracking." In Proc. AIED 2015.

132. Rosalind W. Picard, Elias Vyzas, and Jennifer Healey. "Toward Machine Emotional Intelligence: Analysis of Affective Physiological State." IEEE Transactions on Pattern Analysis and Machine Intelligence 23(10), 2001.

133. V. Podgorelec, and S. Kuhar. "Taking advantage of education data: Advanced data analysis and reporting in virtual learning environments." Elektronika ir Elektrotechnika 114.8 (2011): 111-116.

134. Ming-Zher Poh, Daniel McDuff, and Rosalind Picard. "A medical mirror for non-contact health monitoring." In ACM SIGGRAPH 2011 Emerging Technologies, 2.

135. Ming-Zher Poh, Daniel J. McDuff, and Rosalind W. Picard. "Non-contact, automated cardiac pulse measurements using video imaging and blind source separation." Optics Express, 18(10), 10762-10774.

136. Dimitri Popolov, Michael Callaghan, and P. Luke. "Tying models of learning to design of collaborative learning software tools." Journal of Computer Assisted Learning 18.1 (2002): 46-47.

137. National Research Council. "A framework for K-12 science education: Practices, crosscutting concepts, and core ideas." National Academies Press, 2012.

138. Lawrence R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

139. Roy Rada. "Collaborative Hypermedia in a Classroom Setting." Journal of Educational Multimedia and Hypermedia 3.1 (1994): 21-36.

140. Zahra Rahimi, Diane J. Litman, Richard Correnti, Lindsay Clare Matsumura, Elaine Wang, and Zahid Kisa. "Automatic scoring of an analytical response-to-text assessment." In International Conference on Intelligent Tutoring Systems, pp. 601-610. Springer International Publishing, 2014.

141. Julie Rico, and Stephen Brewster. "Usable gestures for mobile interfaces: evaluating social acceptability." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 887-896. ACM, 2010.

142. Evan F. Risko, Dawn Buchanan, Srdan Medimorec, and Alan Kingstone. "Everyday attention: mind wandering and computer use during lectures." Computers & Education 68 (2013): 275-283.

143.   Ralph Robinson. "Calibrated Peer Review™: an application to increase student reading & writing skills." The American Biology Teacher 63.7 (2001): 474-480.

144.   Eleanor Rosch, and Carolyn B. Mervis. "Family resemblances: Studies in the internal structure of categories." Cognitive Psychology, 7(4), pp. 573-605. Elsevier, Amsterdam (1975)

145.   François Rousseau, Emmanouil Kiagias, and Michalis Vazirgiannis. "Text categorization as a graph classification problem." In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 1702–1712.

146.   Dennis W. Rowe, John Sibert, and Don Irwin. "Heart Rate Variability: Indicator of User State as an Aid to Human-Computer Interaction." In Proc. CHI 1998.

147.   Christopher G. Scully, Jinseok Lee, Joseph Meyer, Alexander M. Gorbach, Domhnull Granquist-Fraser, Yitzhak Mendelson, and Ki H. Chon. "Physiological parameter monitoring from optical recordings with a mobile phone." Biomedical Engineering, IEEE Transactions on, 59(2), 303-306.

148.   Fulvia Seccareccia, Fabio Pannozzo, Francesco Dima, Anna Minoprio, Antonio Menditto, Cinzia Lo Noce, and Simona Giampaoli. "Heart rate as a predictor of mortality: the MATISS project." American Journal of Public Health, 91(8), 1258-1263.

149.   Jinwook Seo, and Ben Shneiderman. "Interactively exploring hierarchical clustering results [gene identification]." Computer, 35(7), 80-86. IEEE, New York (2002)

150.   Dhawal Shah. "MOOCs in 2014: Breaking Down the Numbers", edSurge 2014.

151.   Xiaonan Susan Shen. (1989). "Toward a register approach in teaching Mandarin tones." Journal of the Chinese Language Teachers Association, 24(3), 27-47.

152.   R. B. Singh. "Heart Rate Measurement Through Photoplethysmography." In Proc. BEATS 2010, 170-174.

153.   Sumit Basu, Chuck Jacobs, and Lucy Vanderwende. "Powergrading: a Clustering Approach to Amplify Human Effort for Short Answer Grading." In Transactions of the ACL (TACL), 1 (October), 2013.

154.   Daniel Szafir, and Bilge Mutlu. "ARTFul: adaptive review technology for flipped learning." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2013.

155.   Matthew JW Thomas. "Learning within incoherent structures: The space of online discussion forums." Journal of Computer Assisted Learning 18.3 (2002): 351-366.

156.   Steven L. Thorne, Rebecca W. Black, and Julie M. Sykes. "Second Language Use, Socialization, and Learning in Internet Interest Communities and Online Gaming." The Modern Language Journal, Volume 93, Issue Supplement s1, pages 802–821 (2009).

157.   Feng Tian, Fei Lv, Jingtao Wang, Hongan Wang, Wencan Luo, Matthew Kam, Vidya Setlur, Guozhong Dai, and John Canny. "Let's play chinese characters: mobile learning approaches via culturally inspired group games." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1603-1612. ACM, 2010.

158.   David Tinapple, Loren Olson, and John Sadauskas. "CritViz: Web-based software supporting peer critique in large creative classrooms." Bulletin of the IEEE Technical Committee on Learning Technology 15.1 (2013): 29.

159.   Keith Topping. "Peer assessment between students in colleges and universities." Review of educational Research 68.3 (1998): 249-276.

160.   Kurt VanLehn. "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems." Educational Psychologist 46.4 (2011): 197-221.

161.   Katrien Verbert, Erik Duval, Joris Klerkx, Sten Govaerts, and José Luis Santos. "Learning analytics dashboard applications." American Behavioral Scientist (2013): 0002764213479363.

162.   Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. "Constrained K-Means clustering with background knowledge." In ICML, vol 1, pp. 577-584. 2001.

163.   Yue Wang, Michelle M. Spence, Allard Jongman, and Joan A. Sereno. (1999). "Training American listeners to perceive Mandarin tones." The Journal of the Acoustical Society of America, 106(6), 3649-3658.

164.   Yue Wang, Allard Jongman, and Joan A. Sereno. (2003). "Acoustic and perceptual evaluation of Mandarin tone productions before and after perceptual training." The Journal of the Acoustical Society of America, 113(2), 1033-1043.

165.   Tom Wells, Justin T. Bailey, and Michael W. Link. "Comparison of smartphone and online computer survey administration." Social Science Computer Review 32.2 (2014): 238-255.

166.   Carolyn M. White. "Tonal perception errors and interference from English intonation." Journal of the Chinese Language Teachers Association 16.2 (1981): 27-56.

167.   Joseph Jay Williams, Tania Lombrozo, Anne Hsu, Bernd Huber, and Juho Kim. "Revising Learner Misconceptions Without Feedback: Prompting for Reflection on Anomalies." Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. ACM, 2016.

168.   Ernst Wit, Edwin van den Heuvel, and Jan-Willem Romeijn. "All models are wrong...: an introduction to model uncertainty." Statistica Neerlandica 66.3 (2012): 217-236.

169.   Beverly Woolf, Winslow Burleson, Ivon Arroyo, Toby Dragon, David Cooper, and Rosalind Picard. "Affect-aware tutors: recognising and responding to student affect." International Journal of Learning Technology 4.3-4 (2009): 129-164.

170.   Xiang Xiao, and Jingtao Wang. "Context and cognitive state triggered interventions for mobile MOOC learning." Proceedings of the 18th ACM International Conference on Multimodal Interaction. ACM, 2016.

171.   Xiang Xiao, and Jingtao Wang. "Towards Attentive, Bi-directional MOOC Learning on Mobile Devices." In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. ACM, 2015.

172.   Xiang Xiao, Teng Han, and Jingtao Wang. "LensGesture: augmenting mobile interactions with back-of-device finger gestures." In Proc. ICMI 2013, 287-294.

173.   Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. "Distance Metric Learning with Application to Clustering with Side-Information." In Advances in neural information processing systems, pp. 505-512 (2002)

174.   Janet Zhiqun Xing. "Teaching and Learning Chinese as a Foreign Language." Electronic Journal of Foreign Language Teaching 5, no. 1 (2008): 174-176.

175.   Wenting Xiong, Diane Litman, Jingtao Wang, and Christian Schunn. "An interactive analytic tool for peer-review exploration." Proceedings of the Seventh Workshop on Building Educational Applications Using NLP. Association for Computational Linguistics, 2012.

176.   Zhen Yue, Eden Litt, Carrie J. Cai, Jeff Stern, Kathy K. Baxter, Zhiwei Guan, Nikhil Sharma, and Guangqiang George Zhang. "Photographing information needs: the role of photos in experience sampling method-style research." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1545-1554. ACM, 2014.