

# Enhancing Instructor-Student and Student-Student Interactions with Mobile Interfaces and Summarization

Wencan Luo, Xiangmin Fan, Muhsin Menekse, Jingtao Wang, Diane Litman

University of Pittsburgh  
Pittsburgh, PA 15260 USA

{wel55, xif14, muhsin, jingtaow, dlitman}@pitt.edu

## Abstract

Educational research has demonstrated that asking students to respond to reflection prompts can increase interaction between instructors and students, which in turn can improve both teaching and learning especially in large classrooms. However, administering an instructor's prompts, collecting the students' responses, and summarizing these responses for both instructors and students is challenging and expensive. To address these challenges, we have developed an application called CourseMIRROR (**M**obile **I**n-situ **R**eflections and **R**eview with **O**ptimized **R**ubrics). CourseMIRROR uses a mobile interface to administer prompts and collect reflective responses for a set of instructor-assigned course lectures. After collection, CourseMIRROR automatically summarizes the reflections with an extractive phrase summarization method, using a clustering algorithm to rank extracted phrases by student coverage. Finally, CourseMIRROR presents the phrase summary to both instructors and students to help them understand the difficulties and misunderstandings encountered.

## 1 Introduction

In recent years, researchers in education have demonstrated the effectiveness of using reflection prompts to improve both instructors' teaching quality and students' learning outcomes in domains such as teacher and science education (Boud et al., 2013; Menekse et al., 2011). However, administering an instructor's prompts, collecting the students' reflective responses, and summarizing these responses for

instructors and students is challenging and expensive, especially for large (e.g., introductory STEM) and online courses (e.g., MOOCs). To address these challenges, we have developed CourseMIRROR, a mobile application<sup>1</sup> for collecting and sharing learners' in-situ reflections in large classrooms. The instant on, always connected ability of mobile devices makes the administration and collection of reflections much easier compared to the use of traditional paper-based methods, while the use of an automatic summarization algorithm provides more timely feedback compared to the use of manual summarization by the course instructor or TA.

From a natural language processing (NLP) perspective, the need in aggregating and displaying reflections in a mobile application has led us to modify traditional summarization methods in two primary ways. First, since the linguistic units of student inputs range from single words to multiple sentences, our summaries are created from extracted phrases rather than from sentences. Phrases are also easy to read and browse, and fit better on small devices when compared to sentences. Second, based on the assumption that concepts (represented as phrases) mentioned by more students should get more instructor attention, the phrase summarization algorithm estimates the number of students semantically covered by each phrase in a summary. The set of phrases in a summary and the associated student coverage estimates are presented to both the instruc-

---

<sup>1</sup>CourseMIRROR homepage: <http://www.coursemirror.com>; free download link in Google Play Store: <https://play.google.com/store/apps/details?id=edu.pitt.cs.mips.coursemirror>

tors and the students to help them understand the difficulties and misunderstandings encountered from lectures.

## 2 Demonstration

One key challenge for both instructors and students in large classes is how to become aware of the difficulties and misunderstandings that students are encountering during lectures. Our demonstration will show how CourseMIRROR can be used to address this problem. First, instructors use the server side interface to configure a set of reflection prompts for an associated lecture schedule. Next, students use the mobile client to submit reflective responses to each assigned prompt according to the schedule. Finally, after each submission deadline, both students and instructors use CourseMIRROR to review an automatically generated summary of the student responses submitted for each prompt. The whole functionality of CourseMIRROR will be demonstrated using the scenario described below. In this scenario, Alice is an instructor teaching an introduction to engineering class and Bob is one of her students.

In order to use CourseMIRROR, Alice first logs in to the server and sets up the lecture schedule and a collection of reflection prompts.

Bob can see all the courses he enrolled in after logging into the CourseMIRROR client application<sup>2</sup>. After selecting a course, he can view all the lectures of that course (Fig. 1.a).

After each lecture, Bob writes and submits reflections through the reflection writing interface (Fig. 1.b). These reflections are transmitted to the server and stored in the database. In order to collect timely and in-situ feedback, CourseMIRROR imposes submission time windows synchronized with the lecture schedule (from the beginning of one lecture to the beginning of the next lecture, indicated by an edit icon shown in Fig. 1.a). In addition, to encourage the students to submit feedback on time, instructors can send reminders via mobile push notifications to the students' devices.

After the reflection collection phase for a given lecture, CourseMIRROR runs a phrase summariza-

tion algorithm on the server side to generate a summary of the reflections for each prompt. In the CourseMIRROR interface, the reflection prompts are highlighted using a green background, and are followed by the set of extracted phrases constituting the summary. The summary algorithm is described in Section 3; the summary length is controlled by a user-defined parameter and was 4 phrases for the example in Fig. 1.c.

For Bob, reading these summaries (Fig. 1.c) is assumed to remind him to recapture the learning content and rethink about it. It allows him to get an overview of the peers' interesting points and confusion points for each lecture. To motivate the students to read the summaries, CourseMIRROR highlights the phrases (by using light-yellow background) that were included or mentioned by the current user. This functionality is enabled by the proposed summarization technique which tracks the source of each phrase in the summary (who delivers it). We hypothesize that highlighting the presence of one's own reflections in the summaries can trigger the students' curiosity to some extent; thus they would be more likely to spend some time on reading the summaries.

For Alice, seeing both text and student coverage estimates in the summaries can help her quickly identify the type and extent of students' misunderstandings and tailor future lectures to meet the needs of students.

## 3 Phrase Summarization

When designing CourseMIRROR's summarization algorithm, we evaluated different alternatives on an engineering course corpus consisting of handwritten student reflections generated in response to instructor prompts at the end of each lecture, along with associated summaries manually generated by the course TA (Menekse et al., 2011). The phrase summarization method that we incorporated into CourseMIRROR achieved significantly better ROUGE scores than baselines including MEAD (Radev et al., 2004), LexRank (Erkan and Radev, 2004), and MMR (Carbonell and Goldstein, 1998). The algorithm involves three stages: candidate phrase extraction, phrase clustering, and phrase ranking by student coverage (i.e., how many students are associated with those phrases).

<sup>2</sup>Only Android client is provided. The iOS version is under development. Non-Android users now can use an isomorphic web client, optimized for mobile browsers.

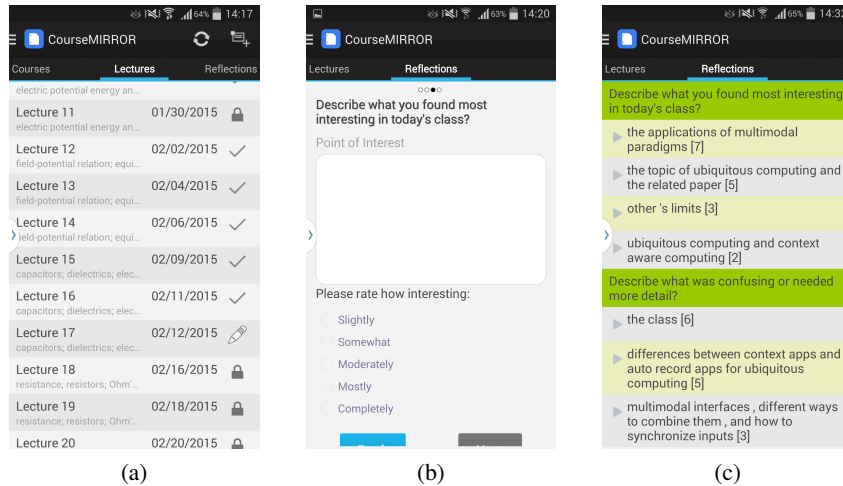


Figure 1: CourseMIRROR main interfaces; a) Lecture list; b) Reflection writing; c) Summary display: the numbers shown in square brackets are the estimated number of students semantically covered by a phrase and a student’s own phrase is highlighted in yellow.

### 3.1 Candidate Phrase Extraction

To normalize the student reflections, we use a parser from the Senna toolkit (Collobert, 2011) to extract noun phrases (NPs) as candidate phrases for summarization. Only NP is considered because all reflection prompts used in our task are asking about “what”, and knowledge concepts are usually represented as NPs. This could be extended to include other phrases if future tasks suggested such a need.

Malformed phrases are excluded based on Marujo et al. (2013) due to the noisy parsers, including single stop words (e.g. “it”, “I”, “we”, “there”) and phrases starting with a punctuation mark (e.g. “t”, “+ indexing”, “?”).

### 3.2 Phrase Clustering

We use a clustering paradigm to estimate the number of students who mention a phrase (Fig. 1.c), which is challenging since different words can be used for the same meaning (i.e. synonym, different word order). We use K-Medoids (Kaufman and Rousseeuw, 1987) for two reasons. First, it works with an arbitrary distance matrix between datapoints. This gives us a chance to try different distance matrices. Since phrases in student responses are sparse (e.g., many appear only once), instead of using frequency-based similarity like cosine, we found it more useful to leverage semantic similarity based on SEMILAR (Rus et al., 2013). Second, it is robust to noise and outliers because it minimizes a sum of pairwise dis-

similarities instead of squared Euclidean distances. Since K-Medoids picks a random set of seeds to initialize as the cluster centers and we prefer phrases in the same cluster are similar to each other, the clustering algorithm runs 100 times and the result with the minimal within-cluster sum of the distances is retained.

For setting the number of clusters without tuning, we adapted the method used in Wan and Yang (2008), by letting  $K = \sqrt{V}$ , where  $K$  is the number of clusters and  $V$  is the number of candidate phrases instead of the number of sentences.

### 3.3 Phrase Ranking

The phrase summaries in CourseMIRROR are ranked by student coverage, with each phrase itself associated with the students who mention it (this enables CourseMIRROR to highlight the phrases that were mentioned by the current user (Fig. 1.c)). In order to estimate the student coverage number, phrases are clustered and phrases possessed by the same cluster tend to be similar. We assume any phrase in a cluster can represent it as a whole and therefore the coverage of a phrase is assumed to be the same as the coverage of a cluster, which is a union of the students covered by each phrase in the cluster. Within a cluster, LexRank (Erkan and Radev, 2004) is used to score the extracted candidate phrases. Only the top ranked phrase in the cluster is added to the output. This process repeats for the next cluster according

to the student coverage until the length limit of the summary is reached.

## 4 Pilot Study

In order to investigate the overall usability and efficacy of CourseMIRROR, we conducted a semester-long deployment in two graduate-level courses (i.e., CS2001 and CS2610) during Fall 2014. These are introductory courses on research methods in Computer Science and on Human Computer Interaction, respectively. 20 participants volunteered for our study; they received \$10 for signing up and installing the application and another \$20 for completing the study. Both courses had 21 lectures open for reflections; 344 reflections were collected overall. We used the same reflection prompts as the study by Menekse et al. (2011), so as to investigate the impact of mobile devices and NLP on experimental results. Here we only focus on interesting findings from an NLP perspective. Findings from a human-computer interaction perspective are reported elsewhere (Fan et al., 2015).

**Reflection Length.** *Students type more words than they write.* The number of words per reflection in both courses using CourseMIRROR is significantly higher compared to the handwritten reflections in Menekse’s study (11.6 vs. 9.7,  $p < 0.0001$  for one course; 10.9 vs. 9.7,  $p < 0.0001$  for the other course) and there is no significant difference between the two CourseMIRROR courses. This result runs counter to our expectation because typing is often slow on small screens. A potential confounding factor might be that participants in our study are Computer Science graduate students while Menekse’s participants are Engineering undergraduates at a different university who had to submit the reflection within a few minutes after the lecture. We are conducting a larger scale controlled experiment (200+ participants) to further verify this finding.<sup>3</sup>

**Questionnaire Ratings.** *Students have positive experiences with CourseMIRROR.* In the closing lecture of each course, participants were given a Likert-scale questionnaire that included two questions related to summarization (“*I often read reflec-*

*tion summaries*” and “*I benefited from reading the reflection summaries*”). Participants reported positive experiences on both their quantitative and qualitative responses. Both questions had modes of 3.7 (on a scale of 1-5,  $\sigma = 0.2$ ). In general, participants felt that they benefited from writing reflections and they enjoyed reading summaries of reflections from classmates. For example, one comment from a free text answer in the questionnaire is “*It’s interesting to see what other people say and that can teach me something that I didn’t pay attention to.*” The participants also like the idea of highlighting their own viewpoints in the summaries (Fig. 1.c). Two example comments are “*I feel excited when I see my words appear in the summary.*” and “*Just curious about whether my points are accepted or not.*”

## 5 Conclusion

Our live demo will introduce CourseMIRROR, a mobile application that leverages mobile interfaces and a phrase summarization technique to facilitate the use of reflection prompts in large classrooms. CourseMIRROR automatically produces and presents summaries of student reflections to both students and instructors, to help them capture the difficulties and misunderstandings encountered from lectures. Summaries are produced using a combination of phrase extraction, phrase clustering and phrase ranking based on student coverage, with the mobile interface highlighting the students’ own viewpoint in the summaries and noting the student coverage of each extracted phrase. A pilot deployment yielded positive quantitative as well as qualitative user feedback across two courses, suggesting the promise of CourseMIRROR for enhancing the instructor-student and student-student interactions. In the future, we will examine how the students’ responses (e.g., response rate, length, quality) relate to student learning performance.

## Acknowledgments

This research is in-part supported by an RDF from the Learning Research and Development Center at the University of Pittsburgh. We also thank all the participants and anonymous reviewers for insightful comments and suggestions.

<sup>3</sup>Due to a currently low response rate, we are also deploying CourseMIRROR in another engineering class where about 50 out of 68 students regularly submit the reflection feedback.

## References

- David Boud, Rosemary Keogh, David Walker, et al. 2013. *Reflection: Turning experience into learning*. Routledge.
- Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336.
- Ronan Collobert. 2011. Deep learning for efficient discriminative parsing. In *International Conference on Artificial Intelligence and Statistics*, number EPFL-CONF-192374.
- Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.
- Xiangmin Fan, Wencan Luo, Muhsin Menekse, Diane Litman, and Jingtao Wang. 2015. CourseMIRROR: Enhancing large classroom instructor-student interactions via mobile interfaces and natural language processing. In *Works-In-Progress of ACM Conference on Human Factors in Computing Systems*. ACM.
- Leonard Kaufman and Peter Rousseeuw. 1987. Clustering by means of medoids. *Statistical Data Analysis Based on the L1-Norm and Related Method*, pages 405–416.
- Luis Marujo, Márcio Viveiros, and João Paulo da Silva Neto. 2013. Keyphrase cloud generation of broadcast news. *arXiv preprint arXiv:1306.4606*.
- Muhsin Menekse, Glenda Stump, Stephen J. Krause, and Michelene T.H. Chi. 2011. The effectiveness of students daily reflections on learning in engineering context. In *Proceedings of the American Society for Engineering Education (ASEE) Annual Conference*.
- Dragomir R. Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Inf. Process. Manage.*, 40(6):919–938, November.
- Vasile Rus, Mihai C Lintean, Rajendra Banjade, Nobal B Niraula, and Dan Stefanescu. 2013. Semilar: The semantic similarity toolkit. In *ACL (Conference System Demonstrations)*, pages 163–168.
- Xiaojun Wan and Jianwu Yang. 2008. Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08.