Identifying Student Misunderstandings using Constructed Responses

Kristin Stephens-Martinez
University of California
Berkeley, USA
ksteph@cs.berkeley.edu

John DeNero
University of California
Berkeley, USA
denero@berkeley.edu

An Ju
Tsinghua University
Beijing, China
ja12@mails.tsinghua.edu.cn

Armando Fox
University of California
Berkeley, USA
fox@cs.berkeley.edu

Colin Schoen
University of California
Berkeley, USA
cschoen@berkeley.edu

L@S 2016, April 25-26, 2016, Edinburgh, Scotland UK
ACM 978-1-4503-3726-7/16/04.
http://dx.doi.org/10.1145/2876034.2893395

Abstract

In contrast to multiple-choice or selected response questions, constructed response questions can result in a wide variety of incorrect responses. However, constructed responses are richer in information. We propose a technique for using each student's constructed responses in order to identify a subset of their stable conceptual misunderstandings. Our approach is designed for courses with so many students that it is infeasible to interpret every distinct wrong answer manually. Instead, we label only the most frequent wrong answers with the misunderstandings that they indicate, then predict the misunderstandings associated with other wrong answers using statistical co-occurrence patterns. This tiered approach leverages a small amount of human labeling effort to seed an automated procedure that identifies misunderstandings in students. Our approach involves much less effort than inspecting all answers, substantially outperforms a baseline that does not take advantage of co-occurrence statistics, proves robust to different course sizes, and generalizes effectively across student cohorts.

Author Keywords

constructed response questions, semi-automatic misunderstanding detection, introductory computer science, education, massive courses

Introduction

In contrast to selected response questions, such as multiple-choice, constructed response questions are much richer in assessing student knowledge and diagnosing student misunderstandings. However in large-enrollment courses, whether online or brick-and-mortar, it is infeasible to expect instructors to read and interpret the constructed responses for all students.

Our goal in this paper is to use these constructed responses at scale to identify misunderstandings in students. We will do this by magnifying a small amount of human labeling effort using trends among the answers from a large cohort of students.

Related Work

Intelligent Tutoring Systems (ITS) use a student model to inform how to help a student. A widely-used approach to create these models is combining an overlay model, in which the student's domain knowledge is represented as a subset of an expert's, with perturbation models, which capture (possibly incorrect) student knowledge not shared by experts [2]. The subset of knowledge representing misconceptions is sometimes called a *bug library*, and originally these were constructed manually by experts.

However, even with automation in creating these bug libraries, it takes much time and effort to create a student model [2]. We propose that by relaxing the desire to create a model that fully and accurately captures student misunderstandings, it is possible to create a practical model that diagnosis common misunderstandings in students without such large effort. Our main goal is the creation of this practical usable artifact for instructors.

Categories for Wrong Answers:

Correct: The answer is actually correct, but marked wrong due to a typo (Ex: `TRue` instead of `True`)

Not an answer (NA): The answer is actually an attempt to do something else, such as quit the interface (Ex: `eixt()`)

Wrong Answer: The answer is actually wrong.

Identifying Misunderstandings in Students

Definition of a Misunderstanding in a Student

A student demonstrates he has a misunderstanding if he responds with two distinct wrong answers that are evidence of that misunderstanding. This definition focuses our analysis on patterns of repeated mistakes.

Labeling Wrong Answers

We have instructors label the top K most common wrong answers (Top-K). To find these, we first remove each student's duplicate (question, wrong-answer) pairs – here on called a *wrong answer*. As a result, we have a series of distinct (student, wrong answer) pairs – here on called *responses*. Then we count the number of responses per wrong answer, sort by this count, and take the top K most common wrong answers across all questions within an assignment. For our evaluation we chose K $= 100$.

We use a two-step process when labeling wrong answers. First, the raters categorize a wrong answer using the categories listed in the margin. Second, for those categorized as wrong are given zero or more *misunderstanding labels*. If a wrong answer has no label, the rater could not determine what misunderstanding could have caused the wrong answer. In addition, an answer may be caused by multiple misunderstandings, hence the allowance for multiple labels.

Using the Label Map

Identifying a misunderstanding in a student involves two steps. Given a student's responses, we first find all labels that are shared by two or more wrong answers. These are identified as misunderstandings in the student.

Second, we propagate labels. A student has additional misunderstanding(s) if three conditions are met: (1) there are two wrong answers where one is not in the Top-K set,

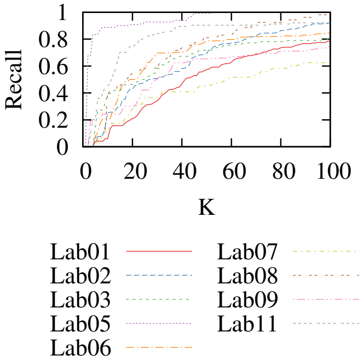

Figure 1: Recall of identified misunderstandings for the fully labeled student sample as K increases.

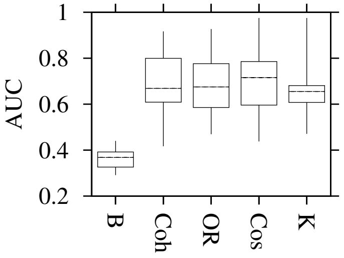

Figure 2: Boxplot of AUC across all labs for baseline and each co-occurrence metric. Note the y-axis does not start at zero.

(2) the wrong answer in Top-K has a misunderstanding label the student does not have, and (3) the two wrong answers have a high co-occurrence.

Case Study: Introductory Computer Science

Empirical Context

We collected wrong answers from UC Berkeley's introductory computer science class's code reading comprehension question sets in nine labs administered by OK [1]. Seven labs had disjoint topics and two were review (Lab03 and 08). Only Lab11 had $< 1,000$ students. The number of wrong answers per student ranged from 0 to 239, with a mean per-lab-median at 14.

Two raters categorized and labeled the Top-K, with K $= 100$. For evaluation, the raters also inspected all wrong answers from a random 50 student sample from each lab – here on called *fully labeled samples*. The category raw inter-rater agreement is 0.92. The fraction of labels given by both raters over all labels given was 0.61. The number of misunderstandings per student in the fully labeled samples ranged from 0 to 15 with a mean per-lab-median of 3.

Given this context, there are ample wrong answers to use and potentially multiple misunderstandings to identify in the students in the fully labeled samples.

Top-100 Exploration

When inspecting what percentage of wrong answers need labeling with K $= 100$, five labs required less than 5%, two required less than 10%, and the last two less than 20%. In other words, for a majority of labs the raters spent 95% less time than if all answers required inspection.

Figure 1 shows recall for identifying misunderstandings in the students in the fully labeled samples as K increases. All labs have recall greater than 0.6 when K $= 100$. This figure shows K $= 100$ is potentially more than enough because marginal returns for each additional wrong answer quickly decrease and is small by the time K $= 100$.

Co-occurrence Metrics Exploration

We considered four co-occurrence metrics: odds ratio, coherence, cosine, and kulcynski, mainly inspired by pattern mining literature [4]. We compared these metrics to an uninformed baseline where if a student has a wrong answer with an unshared label that student receives the label with a probability of p.

To compare the metrics and the baseline we chose to use the area under the curve (AUC). However, the high recall when K $= 100$ and 1.0 precision by our definition of identifying misunderstandings in students needs to be handled. For easier comparison we anchored each graph at the baseline's extreme points, which are the same for all, and scaled the curve such that it fits inside a 1x1 square.

Figure 2 shows the AUC values across all labs for the baseline and metrics using a boxplot with quartiles. All metrics perform better than the baseline, except when Lab11's baseline AUC is higher than its coherence. These results show that the co-occurrence metric is providing useful information in propagating labels. However, which metric performs best is unclear.

Scaling and Stability of Top-K

To understand the scale of how many students are actually needed to achieve good recall for each lab and the stability of the Top-K wrong answers, we investigated the recall on the fully labeled sample and the overlap with the overall Top-100 for increasing values of N sampled students. We ran each N student sample 50 times and plotted the mean recall and percent overlap in Figure 3.

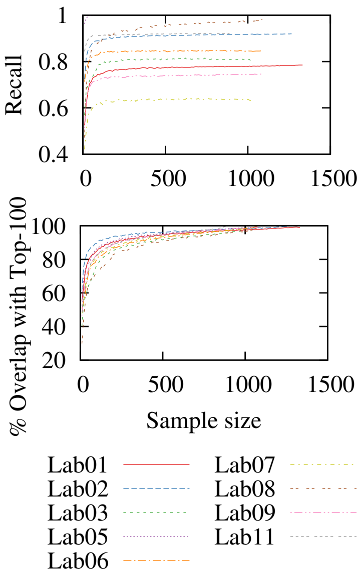

Figure 3: Comparison of recall with fully labeled samples and % of overlap with overall Top-100 for increasing samples of the students. Note neither y-axis starts at 0.

This figure shows that the number of students needed for reasonable scale and stability is only a few hundred.

Future Work

Co-Occurrence Metric Investigation for Propagation
While all metrics performed better than baseline, there was no clear best metric to use. We plan to investigate further in both which metrics to use and how to use these metrics to determine if a label should be propagated.

System Deployment for Guidance Messaging
To better understand the misunderstanding diagnostic model's performance, we plan to deploy it in OK. We will use it to decide what guidance message to give a student to help him learn the misunderstood concept.

In addition, we will investigate what kind of guidance best helps computer science learning and what factors need consideration. Education research has found many kinds of effective guidance messages and other factors which influencing message effectiveness [3]. We plan to compare concept reteaching and knowledge integration guidance, as well as investigate how message timing and student prior coding experience and achievement level influence the guidance's effectiveness.

Conclusion

Prior work expends much effort on creating student models to then identify misunderstandings. We propose a more practical way by relaxing the need to identify all misunderstandings to at least the common ones. Our empirical technique expends less effort than creating a student model, while successfully diagnosing misunderstandings in students.

Our model is based on instructors inspecting the top common wrong answers by categorizing and labeling them with associated misunderstandings. Then, using this mapping from wrong answers to labels, we identify misunderstandings in students by both using only the map and propagating labels using co-occurrence metrics. In our introductory computer science class case study, we inspected the top 100 most popular wrong answers. We achieved at least 0.6 recall and perfect precision across all nine lab assignments and when including propagation all metrics under consideration, in all cases but one, performed better than a random baseline.

Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1106400.

References

[1] Basu, S., Wu, A., Hou, B., and DeNero, J. Problems before solutions: Automated problem clarification at scale. In *Proceedings of the Second ACM Conference on Learning@ Scale* (2015), 205–213.

[2] Holt, P., Dubs, S., Jones, M., and Greer, J. The state of student modelling. In *Student modelling: The key to individualized knowledge-based instruction*. Springer, 1994, 3–35.

[3] Shute, V. J. Focus on formative feedback. *Review of Educational Research* (2008), 153–189.

[4] Wu, T., Chen, Y., and Han, J. Re-examination of interestingness measures in pattern mining: a unified framework. *Data Mining and Knowledge Discovery* (2010), 371–397.