Towards Crowdsourcing the Identification of Knowledge Components

Steven Moore
Carnegie Mellon University
Pittsburgh, United States
StevenJamesMoore@gmail.com

Huy A. Nguyen
Carnegie Mellon University
Pittsburgh, United States
hn1@andrew.cmu.edu

John Stamper
Carnegie Mellon University
Pittsburgh, United States
john@stamper.org

ABSTRACT
Assigning a set of hypothesized knowledge components (KCs) to assessment items within an ed-tech system enables us to better estimate student learning. However, creating and assigning these KCs is a time-consuming process that often requires domain expertise. In this study, we present the results of crowdsourcing KCs for problems in the domain of mathematics and English writing, as a first step in leveraging the crowd to expedite this task. Crowdworkers were presented with a problem and asked to provide the underlying skills required to solve it. Additionally, we investigated the effect of priming crowdworkers with related content before having them generate these KCs. We then analyzed their contributions through qualitative coding and found that across both the math and writing domains roughly 33% of the crowdsourced KCs directly matched those generated by domain experts for the same problems.

Author Keywords
Knowledge component; Knowledge component modeling; Crowdsourcing; Expertise; Courseware improvement.

CSS Concepts
• **Human-centered computing~Collaborative and social computing** • Human-centered computing~Social tagging systems • Computing methodologies~Cognitive science

INTRODUCTION
Many educational technologies, such as intelligent tutoring systems and online courseware, utilize knowledge component modeling to support their adaptivity. This treats student knowledge as a set of interrelated KCs, where each KC is "an acquired unit of cognitive function or structure that can be inferred from performance on a set of related tasks" [3]. A KC model is defined as a mapping between each question item and a hypothesized set of associated KCs that represent the skills or knowledge needed to solve that item. This mapping is intended to capture the student's underlying cognitive process and is vital to many core functionalities of an intelligent educational software, enabling features such as adaptive feedback and hints [7].

The construction of such a mapping is typically carried out by learning science practitioners, such as subject matter experts, cognitive scientists and learning engineers, who inspect the materials and assign one or more KCs to each question [4]. This process is often a time-consuming task, making both the creation of this map and continuous iteration challenging. An emerging area that has the potential to provide the human resources needed for scaling KC modeling is crowdsourcing [6]. Naturally, the challenge with this approach is that the population of crowdworkers is highly varied in their education level and domain knowledge proficiency [8]. One method to potentially address this issue is to prime crowdworkers, having them solve several related problems before engaging in the KC modeling. Priming is done through exposing an individual to content before a certain problem, which can help them have an easier time recalling the concepts needed for the task at hand [1]. In this study, we would like to see if priming can be helpful in a crowdsourcing context instead of in the traditional learning context.

Therefore, as a first step towards examining and promoting the feasibility of crowdsourced KC modeling, we studied how crowdworkers can provide the underlying KCs for an assessment activity. Using a crowdsourcing platform, we gathered participants and asked them to provide three KCs needed to solve a given problem in the domains of math and English writing, with or without having solved some related priming questions. Our research questions are as follows: **RQ1**: *Are crowdworkers able to generate knowledge components that domain experts have identified?* **RQ2**: *Does priming participants improve the quality and quantity of the knowledge components they generate?*

From these questions, our goal is to see whether it is possible to employ crowdsourcing to generate a baseline KC model that is both interpretable and able to be translated into a more learnersourced context.

METHODS
The study consists of two related experiments that differ in their domain content for a specific part of the task. The first domain is mathematics, with a focus on the area of shapes, such as squares and rectangles. The second domain is English writing, with a focus on prose style involving agents and

L@S'20, August 12–14, 2020, Virtual Event, USA.

ACM ISBN 978-1-4503-7951-9/20/08...$15.00.
DOI: https://doi.org/10.1145/3386527.3405940

clause topics. The math content is at the middle school level and the writing content at the undergraduate level in the United States. For both of these domains, we conducted an experiment using the Amazon Mechanical Turk (AMT) platform. Eighty unique crowd workers on AMT, known as turkers, completed the math experiment and sixty unique turkers completed the writing, for a total of 140 participants. Thirteen participants in the Writing experiment were removed from our analyses due to submitting invalid responses that indicated either a misunderstanding of the experiment instructions or behavior similar to a bot. Filtering these invalid participants left us with 47 total participants in writing, combined with the 80 from math, for a total of 127 participants. Among these participants, 55 self-identified as female and 72 as male. The mean selected age range was 35-44. All participants reported having at least a high school degree or equivalent. In each experiment, the tasks took roughly five minutes to complete. Participants were compensated $0.75 upon completion, providing a mean hourly wage of $9.

Each turker was assigned to one of two conditions. In the priming condition, participants solved two problems that cover content related to the area of different figures in the math experiment, and clause/actions of different sentences in the writing experiment; these problems were intended to prime participants for the subsequent main task. In the no-priming condition, these problems were not included, and participants moved straight to the main task.

In the main task, participants were given a word problem and asked to list three KCs that are required to solve the problem. The prompt for KCs was "*As concisely as possible, please indicate three skills required to answer the above math problem about the wall.*" in the math experiment and "*As concisely as possible, please indicate a three skills required to answer the above question that involved revising the sentence.*" in the writing experiment. Note that the prompt uses the term "skill" rather than "knowledge component" to avoid jargon that may be confusing.

Math and Writing Conditions

The word problem for which participants were asked to generate KCs for in the math domain comes from a middle school algebra intelligent tutoring system. It involves finding the area of a wall that has a door and windows embedded within it as composite shapes. The problem in the writing domain comes from an online prose style course for freshmen and sophomores at a four-year university in the United States. It asks students to revise a sentence, so that the agent is in the subject position. An expert math instructor familiar with the domain knowledge and KC modeling process tagged the math problem with three KCs as described in Table 1. Similarly, a different expert instructor who taught the online writing course from which the problem comes from provided the four KCs for it, also in Table 1. These expert-generated KCs will serve as a baseline for our comparison with the turkers' generated KCs.

KC (*Domain*)	Definition
Compose-by-addition (*M*)	In an equation such as a + b = c, given any two of a, b or c, find the third variable.
Subtract (*M*)	Subtract the area of one shape from another.
Rectangle-area (*M*)	Finding the area of a rectangle shape.
Id-clause (W)	Identify the clause-level topic of a sentence.
Discourse-level (W)	Keep the discourse-level topic of the sentence in focus
Subject-position (W)	Assess whether an entity is a subject
Verb-form (W)	Transform a passive verb to active verb

Table 1. Expert-generated KCs in the math and writing experiment. The domain code "M" stands for math and "W" for writing.

Coding of Knowledge Components

Once all the participants completed the task, we manually coded their 381 responses (240 from math, 141 from writing). We considered a participant's full text response to each of the three skill inputs in the main task as one *contributed KC,* so in total each participant made three contributions. Overall, the responses mostly consisted of sentence fragments and the occasional full sentences. No participant contributed any text longer than one full sentence, which is in line with the conciseness requirement in the provided instructions. With each input being treated as a single unit, a codebook was developed and iteratively refined between two research assistants. The codebook was then applied to the full dataset from each domain by the two research assistants, akin to [2]. Next, the code agreement was measured via Inter-Rater Reliability (IRR). The coders achieved a Cohen's kappa of $\kappa = 0.924$ in the math experiment and $\kappa = 0.901$ in the writing experiment, both indicating a high level of agreement [5].

With the codes applied, the contributed KCs were then categorized into three groups based on the extent of their applicability to the problems and whether they matched the expert-generated KCs. The first category is for responses that are specific to the problem and offer enough detail to indicate a skill that might technically be needed to solve it. We refer to such explanations as *Relevant* henceforth. Our second category, *Direct Match*, is representative of contributed KCs whose descriptions align with those of the expert-generated KCs. Note that this is a subset of *Relevant*, i.e., any Direct Match contribution is also Relevant. The third category denotes responses that are too broad to fit the problems, too vague, or do not convey a clear meaning. These types of responses are considered *Irrelevant* contributions.

RESULTS

RQ1: *Are crowdworkers able to generate knowledge components that domain experts have identified?* The math experiment yielded 240 responses in total from both

conditions. In the priming condition, 87/120 (72.5%) contributed KCs were categorized as *Relevant* and 38/120 (31.67%) were *Direct Match*. In the no-priming condition, 94/120 (78.33%) were categorized as *Relevant*, and 38/120 (31.67%) were also *Direct Match*. In other words, roughly one third of the KCs in the math experiment were equivalent to those from domain experts.

The writing experiment yielded 141 responses in total from both conditions. In the priming condition, 39/66 (59.09%) contributed KCs were categorized as *Relevant* and 24/66 (36.36%) were *Direct Match*. In the no-priming condition, 41/75 (54.67%) contributed KCs were categorized as *Relevant* and 26/75 (34.67%) were *Direct Match*. Table 2 shows a breakdown of how many of the expert-generated KCs were matched by participants in each domain and condition.

KC (*Domain*)	Priming	No Priming
Compose-by-addition (M)	0	0
Subtract (M)	26	24
Rectangle-area (M)	12	14
Id-clause (W)	0	0
Discourse-level (W)	4	8
Subject-position (W)	11	6
Verb-form (W)	9	12

Table 2. Count of contributed KCs in each condition that matched the expert-generated KCs. The domain code "M" stands for math and "W" for writing.

RQ2: *Does priming participants improve the quality and quantity of the knowledge components they generate?* First, we looked at the contributed KCs that are *Relevant* (Table 3). An unpaired t-test showed no significant difference in the number of *Relevant* KCs between the two conditions, $t(77) = -.778$, $p = .439$. Similar results were observed in the writing condition, where there was no significant difference between the priming and no-priming condition, $t(44) = .366$, $p = .717$.

	Math - Priming	Math – No Priming	Writing - Priming	Writing - No Priming
Relevant M (*SD*)	2.18 (1.06)	2.35 (0.95)	1.77 (1.23)	1.64 (1.25)
Direct Match M (*SD*)	0.95 (0.783)	0.95 (0.749)	1.09 (1.11)	1.04 (1.10)

Table 3. Descriptive statistics of the number of *Relevant* and *Direct Match* KCs in each condition.

However, when comparing between domains (and collapsing the conditions), an independent two-tailed t-test showed that participants in the math domain generated significantly more *Relevant* KCs than those in the writing domain, $t(125) = 2.79$, $p = .006$. We also examined the number of *Relevant* KCs that everyone contributed. Table 4 shows the number of participants who contributed 0, 1, 2 and 3 relevant KCs in each condition; in all cases, most participants had all three responses marked as *Relevant*.

# of Relevant	Math - Priming	Math - No Priming	Writing - Priming	Writing - No Priming
0	5	3	5	7
1	4	4	4	4
2	10	9	4	5
3	21	24	9	9

Table 4. The number of participant contributions categorized as *Relevant* across each condition.

Next, we looked at the contributed KCs that are *Direct Match* (Table 5). An unpaired t-test showed no difference in the number of *Direct Match* KCs between the priming condition and no-priming condition in the math experiment -- $t(77) = 0$, $p = .998$ -- as well as the writing experiment -- $t(44) = .158$, $p = .875$. In other words, having participants do priming problems in the domain of math or writing before the KC labeling task had no effect on the number of contributions that were categorized as *Direct Match*. Furthermore, unlike in the case of *Relevant* KCs, there was no significant effect of the domain on the number of *Direct Match* KCs, $t(125) = -.690$, $p = .491$.

# of Direct Match	Math - Priming	Math - No Priming	Writing - Priming	Writing - No Priming
0	10	11	10	11
1	25	21	2	5
2	2	7	8	6
3	3	1	2	3

Table 5. The number of participant contributions categorized as *Direct Match* across each condition.

DISCUSSION AND FUTURE WORK

From our study, we found that crowd workers can be leveraged to identify a subset of knowledge components that comprise a problem. In both the math and writing domain, participants were able to make contributions that were relevant to the problems and, in some cases, even directly matched the KCs generated by expert instructors. Furthermore, we showed that prior training in KC modeling or priming for the problem content was not a necessary prerequisite. While our findings suggest a step towards leveraging crowdworkers to assist with this task, it still required human oversight and some expert-generated KCs were missed by participants altogether. Further work remains to find ways to better leverage participants in both their accuracy of KCs and requiring less processing.

MTurk participants can generate a few accurate KCs

Across both conditions in the math experiment, roughly three-fourths of participant contributions were categorized as relevant. These were akin to KCs and pertinent to the problem, just not at the appropriate level for the problem in the context of the course and for the specific level of the

learner. Around a third of the contributed KCs in the math experiment were also categorized as a direct match to the expert-generated KCs originally tagged to the problems. While some of these were more detailed than others, they were indicative of the KCs being measured by the problems. These *Direct Match* contributions could directly be applied to a skill mapping or underlying KC model used in the system that the problems are hosted in.

In the writing experiment, half of the contributed KCs in each condition were considered relevant. Given that the average education level among our turkers is at the high school level while this domain is at the undergraduate level, we believe this proportion of relevant contributions is an encouraging result. Furthermore, participants in both conditions had a third of their contributions categorized as a direct match to the expert-generated ones. These KCs were often more detailed than the ones provided in the math condition and could also be directly used in the underlying model of a system.

On the other hand, there were no contributed KCs that directly matched the *Compose-by-addition* KC in the math experiment or the *Id-clause* in the writing experiment (Table 2). Based on Table 1, we found that these two KCs have very specific definitions that the average turkers could not be reasonably expected to come up with, especially given that our instruction prompted for conciseness. However, an interesting future work would be designing the prompts in a way that promotes deeper thinking and helps participants get closer to these specific expert-generated KCs while avoiding the mentioned pitfall. Overall, one-third of participant KCs fitting that of experts is not a bad first attempt, but a higher accuracy with less processing is desirable in order to be implemented into an actual system.

Priming participants had little effect on their contributions

In both experiments, we wanted to see how priming participants with related problems before having them perform the KC task might impact their contributions. This was conducted to partially mimic how students might engage in the task during one of their courses, as they would be readily solving such problems. Ultimately there was no effect of priming on the number of relevant or direct match contributions in both the math and writing experiments. In the math experiment, the priming questions were correctly answered by almost all participants, suggesting such questions might have been too easy to engage them in deeper processing. The opposite was observed in the writing experiment, where the priming questions were mostly incorrectly answered, indicating they may have been too difficult. Additionally, since there were only two additional problems, the material may not have been sufficient to prime participants to the content.

While the analyses we performed here may end up taking more time than the task of generating the KCs themselves, we plan to work towards a method that can greatly expedite such tasks. Ultimately, we would like to scale this up into a learnersourced context, such as embedding these prompts into an online course. Our next step is to reduce or automate the need for this heavy qualitative analysis of such contributions if possible, so that the insights provided from the crowdworkers or learners can be better leveraged.

REFERENCES

[1] Barbara A. Dosher and Glenda Rosedale. 1989. Integrated retrieval cues as a mechanism for priming in retrieval from memory. *Journal of Experimental Psychology: General* 118, 2: 191.

[2] Victoria Elliott. 2018. Thinking about the coding process in qualitative data analysis. *The Qualitative Report* 23, 11: 2850–2861.

[3] Kenneth R. Koedinger, Albert T. Corbett, and Charles Perfetti. 2012. The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science* 36, 5: 757–798.

[4] Kenneth R. Koedinger, John C. Stamper, Elizabeth A. McLaughlin, and Tristan Nixon. 2013. Using data-driven discovery of better student models to improve student learning. In *International Conference on Artificial Intelligence in Education*, 421–430.

[5] J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33, 1: 159–174. https://doi.org/10.2307/2529310

[6] Steven Moore, Huy A Nguyen, and John Stamper. 2020. Evaluating Crowdsourcing and Topic Modeling in Generating Knowledge Components from Explanations. In *International Conference on Artificial Intelligence in Education*.

[7] Steven Moore and John Stamper. 2019. Decision support for an adversarial game environment using automatic hint generation. In *International Conference on Intelligent Tutoring Systems*, 82–88.

[8] Donna Vakharia and Matthew Lease. 2015. Beyond Mechanical Turk: An analysis of paid crowd work platforms. *Proceedings of the iConference, 1-17.*