An Explanation-Based Classroom Response System for Real-Time Analysis of Undergraduate Students' Natural Language Explanations

Jordan Esiason[1], Priyanka Khare[1], Claire Aguiar[1], Dan Carpenter[1], Wookhee Min[1], Seung Lee[1], Gamze Ozogul[2], Xiaoying Zheng[2], James Lester[1]

[1]North Carolina State University, Raleigh, NC 27695 USA
[2]Indiana University Bloomington, 107 S. Indiana Avenue, Bloomington, IN 47405-7000
jesiaso@ncsu.edu, pkhare@ncsu.edu, cmaguiar@ncsu.edu, dan.d.carpenter@gmail.com, wmin@ncsu.edu, sylee@ncsu.edu, gozogul@iu.edu, zheng12@iu.edu, lester@ncsu.edu

Abstract

Effective classroom teaching requires instructors to be responsive to their students, such as by pivoting their lectures in real-time to address common misconceptions that their students may have developed. Classroom response systems such as multiple-choice "clicker" systems are one method by which instructors can gauge their students' understanding during classroom lectures, but open-ended questions that prompt students to engage in self-explanation are better suited to promoting critical thinking. Additionally, analyzing students' natural language responses typically requires time-consuming manual analysis, which makes it challenging to implement in a classroom setting. To address this challenge, we present an LLM-driven method for automatically assessing students' responses and generating an aggregated summary of LLM-based evaluations for their self-explanations during undergraduate classroom lectures. Our approach extracts relevant knowledge components for a given question, tags students' responses according to whether they correctly address each knowledge component, and generates class-level summaries that highlight common misconceptions and gaps in knowledge to support instructors in pivoting their lectures in real time. We evaluate the system's effectiveness at these tagging and summarization tasks on data from an undergraduate computer science course, using quantitative and qualitative metrics such as relevance, sufficiency, hallucination rate, and alignment with instructional goals and desired feedback format gathered through instructor interviews. Results suggest that the explanation-based classroom response system can accurately analyze students' natural language explanations.

Introduction

Student interaction and engagement are vital to learning in an undergraduate setting (Beauchamp and Kennewell 2010; Blasco-Arcas et al. 2013). However, some barriers to productive interactivity exist in the typical undergraduate classroom. These include limits on student-instructor interaction due to class sizes, mode of interaction, and other factors (Blasco-Arcas et al. 2013; Stains et al. 2018). "Clicker" style systems allow students to respond to instructors' questions during a lecture and can provide some insight into the class's current knowledge, but they are limited to multiple-choice or fill-in-the-blank questions.

To address this challenge, we present an LLM-based approach for automatically assessing students' responses to open-ended questions and providing class-level feedback summaries to instructors during a lecture while promoting constructive and active learning through self-explanation. Our approach tags student responses with relevant knowledge components (KCs), which are specific pieces of information that contributes to a student's understanding of a broader concept, providing a mechanism for enforcing consistency of grading and enabling instructors to receive real-time summarizations of students' grasp of key concepts. This can enable instructors to examine different student mistakes as identified by the presence or absence of different KC tags.

Previous work has investigated LLMs for similar tasks reliant on KC generation (Moon et al. 2025; Moore et al. 2024; Wei, Carvalho, and Stamper 2025), however the use of these LLM tools faces several hurdles. KC tags must be relevant, correct, and consistently applied across students to ensure fairness in a probabilistic grading environment. Additionally, LLM-generated content may be prone to hallucinations, which could negatively impact students (Kerslake et al. 2025). Finally, each stage of the system must be observable so that instructors can trust but verify LLM-assigned KCs, taggings, and summaries.

We address these issues in this paper by evaluating the performance of a prototype LLM-powered summarization pipeline. This pipeline first attempts to generate a set of KCs based upon a given question produced by an instructor. Then, the system applies KC tags and a short explanation of the tagging to students' responses to short open-response questions posed during an undergraduate computer science course lecture. Lastly, it delivers a written overall summary of class performance on a topic, counts of tags across student responses, and sample student responses for each error label generated. Our evaluation seeks to answer the following research questions about this prototype pipeline's performance using an out-of-the box LLM:

1. Are the generated KCs for each of the instructor's questions relevant, within the scope of the question, and sufficient to identify a correct response?

2. How accurately are the KC tags applied to student responses?
3. Are the summaries generated from instructor questions and their related KC taggings of student responses relevant, correct, and generated in a desirable format as identified by teacher interviews?

Related Work

Interactivity in Learning

Research on the effects of interactivity in learning tends to categorize the types of interactivity in classrooms along dimensions of the control or interactivity that a student has. In Beauchamp et al., a distinction is drawn between settings that offer no or little control over learning and question response (didactic) and those that offer the opportunity to engage in more flexible and interactive learning experiences (dialogic) (Beauchamp and Kennewell 2010).

Traditional large undergraduate lecture-style settings are the most common undergraduate educational setting (Grissom, Mccauley, and Murphy 2017; Stains et al. 2018) but typically offer little to no control or didactic learning experiences where a lecturer guides students through content (Beauchamp and Kennewell 2010; Blasco-Arcas et al. 2013). This has been found to be problematic, as higher levels of interactivity have been widely demonstrated to generally encourage greater positive learning outcomes and engagement (Beauchamp and Kennewell 2010; Das and Lim 2024; Dbiec 2018; Lim 2017; Petrović and Pale 2021; Renaud and Cutts 2013; Wang et al. 2021; Williams et al. 2011). Such benefits can be seen in learning environments such as those enhanced with classroom response systems (CRS), which present better learning opportunities.

Classroom Response Systems

A wide range of interactive technologies are available to teachers in the classroom, but clicker-style CRS that augment lectures with short multiple-choice activities have been one of the most popular (Milo D. Koretsky and Higgins 2016). These systems have been shown to improve engagement (Dong et al. 2017; Hung 2016) and reasoning (Milo D. Koretsky and Higgins 2016) over traditional lecture-style settings. However, other work has generally found small (or sometimes null) advantages to clicker use with regards to learning outcomes (Hubbard and Couch 2018; Hunsu, Adesope, and Bayly 2016; Liu et al. 2017; Pisheh et al. 2018). Additionally, some studies suggest that improved learning outcomes with CRS may preferentially occur in students that are already high-performing (Hubbard and Couch 2018), indicating that typical multiple-choice CRS may be primarily beneficial to students that are predisposed to do well in their courses to begin with.

Other benefits and uses of CRS outside of cognitive outcomes and engagement have been studied as well. One study examined the impacts of CRS on anxiety in students and found that students reported lower anxiety when participating in CRS exercises as compared to other modes of participation, such as being volunteered for a response during lecture by an instructor (England, Brigati, and Schussler 2017). CRS can also be readily extended to serve as data collection instruments for tasks they were not originally designed for such as predicting at-risk students based on CRS performance (Choi et al. 2018). In addition, they have been used to create adaptive homework systems (Rodríguez-Martínez et al. 2022) and to provide feedback (Papadopoulos et al. 2018).

Open-response-based systems may provide additional benefit to learning through self-explanation as compared to relatively low-interactivity environments such as multiple-choice clicker-style CRS (Chi and Wylie 2014). There is a large body of evidence to suggest that self-explanation is a powerful tool for enhancing learning outcomes in a variety of settings (Chi and Wylie 2014; Garces et al. 2019), as this process encourages students to engage with material on a deeper level than selecting options out of a multiple-choice lineup. However, automatically tagging and summarizing text responses accurately during a lecture presents a significant challenge to implementing this process.

Automated Knowledge Component Extraction

The derivation and use of KCs is a widely used method of topic modeling and knowledge tracing in the evaluation of question content and student answers. However, much of the existing literature in computer science education focuses on deriving KCs from code exercises rather than open-response questions, while KC extraction as a whole tends to emphasize extracting KCs in a post-hoc manner that requires existing data, over time rather than in single-instance question deployments, or from multiple-choice-style questions (Duan et al. 2025; Moon et al. 2025; Moore et al. 2024; Oliveira Moraes and Pedreira 2021; Shi et al. 2023; Wei, Carvalho, and Stamper 2025). Relatively little research has been conducted on tagging computer science knowledge components in short, instructor-generated open-response questions in a setting where students are expected to give very brief responses, often containing colloquial language and grammar errors. LLMs have shown to be a promising avenue for KC generation by combining the implicit encoding of knowledge in embeddings with powerful text interpretation mechanisms (Duan et al. 2025; Moon et al. 2025; Moore et al. 2024; Wei, Carvalho, and Stamper 2025).

Large Language Models

LLMs have proven to be exceptional tools for tasks such as tagging and document summarization due to the implicit encoding of knowledge in their representations, but some problems persist in their implementation and must be accounted for. Due to their probabilistic nature, LLMs are prone to hallucination issues. These hallucinations pose substantial risks in fields like education, where LLMs can mislead students or harm their confidence in themselves or the LLM (Ji et al. 2023; Kerslake et al. 2025; Kirstein et al. 2025). Kirstein et al. (Kirstein et al. 2025) point out that while some problems in the natural language processing field such as topic flow and content repetition in text are mostly mitigated, other problems such as idiosyncrasies in informal language, comprehension, and factuality persist. All of these issues are

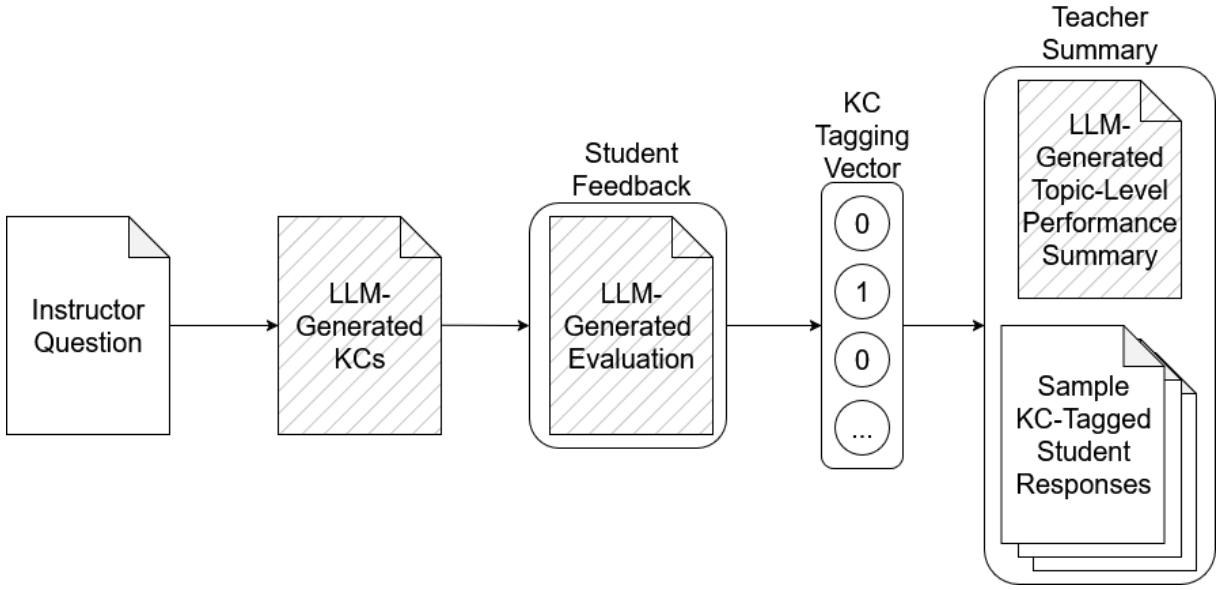

Figure 1: The data processing pipeline.

likely to be present in the task of tagging and summarization of short text responses produced by students during lectures.

The ExplainIt Classroom Response System

ExplainIt is a multi-stage CRS that allows instructors to prepare short open-response questions, present them to students during lectures, gather responses from students, and provide automated feedback on those responses. Instructors prepare questions ahead of time, and then present them to students who respond on a laptop or hand-held device of their choice. This engages students in self-explanation, allowing for deeper learning than pure lecturing or other CRSs. LLM-driven, immediate, personalized feedback then aids students in troubleshooting their response. Instructors are also presented with a real-time analytical dashboard of student responses, which we propose to extend with KC tagging and feedback summaries in this work. An important design principle of ExplainIt is to create a low-overhead tool for instructors to use such that integration of the tool provides benefits without adding to their workload. This has informed the heavily automated design choices underlying this system.

Methods

Data Collection

ExplainIt was deployed to 50 students in one undergraduate computer science course on applied learning and data analytics. Students used ExplainIt for 33 days over the Fall semester of 2024. Students were prompted by the instructor to use the ExplainIt CRS to respond to 27 unique open-response questions, resulting in 1,322 student responses across all questions. The questions posed to the students and used in this analysis are shown in Figure 5. Not all students persisted throughout the entire course, resulting in later questions having 48 rather than 50 responses.

KCs, KC-tagged feedback, and summaries were generated with the GPT-4o mini model using the OpenAI API (OpenAI 2023, 2025). A pipeline was created to parse and summarize instructor questions and student responses as shown in Figure 1. First, the LLM was prompted to derive KCs from each instructor question as shown in Figure 2. Then, the questions and their corresponding KCs were used to generate a list of Boolean values indicating whether or not the language model flagged the presence or absence of each individual KC and a short rationale as shown in Figure 3. Lastly, each question, their respective Boolean lists, and any generated explanations of the lists were summarized by the language model into a format intended for instructor review, informed by two teacher interviews, as shown in Figure 4.

Inter-Rater Reliability

Inter-rater reliability (IRR) was established on the generated KCs and summarization datasets using Gwet's AC1. AC1 was chosen due to its lack of sensitivity to trait prevalence and marginal probabilities as compared to other mea-

> An instructor posed the following question during an undergraduate computer science lecture: {question}
>
> Create a list of knowledge components covered by an ideal response to this question. Your answer should consist of ONLY the list of knowledge components described in a few words each.

Figure 2: The prompt used for generating knowledge components. The input is the instructor's question, and the output is an easily parsed list of knowledge components.

An instructor posed the following question during an undergraduate computer science lecture: {question}

A student provided the following response to the question: {response}

Identify whether or not the following numbered knowledge components are present in the student's response: {KCs}

Your response should consist of the above list of knowledge components with "True" if the component is present, or "False" if it is not. Additionally, provide an overall evaluation of the students response and a grade of "Correct" if the student included several key knowledge components, "Partially Correct" if the student only included insufficient knowledge components to demonstrate reasonable understanding, or "Incorrect" if the student response is completely irrelevant or included too few knowledge components to demonstrate even partial understanding. Use no more than a few sentences in your evaluation.

Figure 3: The prompt used for tagging a student response with knowledge components. The input is the relevant generated KCs for an instructor's question and the student's response, and the output is an easily-parsed list of Boolean tags for the presence or absence of each knowledge component.

sures, such as Cohen's Kappa or Fleiss' generalized Kappa (Gwet 2002). AC1 has been used in evaluations of IRR in the machine learning and LLM fields of research (Bewersdorff et al. 2023; Touvron et al. 2023). This was deemed important both due to the small size of these two datasets (only 27 artifacts each were produced for the generated knowledge component sets and summaries) and also because certain traits, such as presence of hallucinations, were expected to be rare due to the increasing reliability of the GPT platform on summarization tasks (OpenAI 2023). Two experts evaluated the data. The KC-tagged student responses were evaluated for accuracy by one expert grader.

Evaluation of generated KCs measured their relevancy to the question, their sufficiency in capturing whether or not a student successfully responded to the question, and whether or not the KC was within the scope of the question. Relevancy was determined by whether or not a generated KC was directly related to the topic of the question. Sufficiency was graded as a matter of expert grader opinion (e.g. "if a student provided a response with all of these KCs, would they be graded as correct?"). Scope was graded as whether or not all generated KCs were required to identify a correct response; for example, a comprehensive definition of k-fold cross validation is not necessary in order to explain the basic steps of the algorithm (Figure 6). Determinations of what was required to identify sufficiency or a correct response were made by a combination of expert grader opinion and reference to the course material.

Instructor question: {question}

Feedback given to individual students: {feedbackList}

Students were asked the given question by an instructor.

Students were assigned a score of 'Correct', 'Partially correct', or 'Incorrect' and provided with the given feedback based on their answers to the question posed.

Summarize this feedback that was given to individual students such that a classroom instructor would be able to understand what concepts relevant to the given question students may be struggling with.
Please be concise in your response.

Figure 4: The prompt used for generating summaries. The input is the instructor's question, a list of Boolean tags generated in the tagging step and their short explanations, and the output is an easily-parsed list of knowledge components.

Evaluation of the KC tags applied to student responses analyzed whether or not the KC was actually present in the student's response based upon expert grader opinion and reference to course materials. Some grading took the form of simple rubrics (for example, establishing the presence of a definition or explanation of a concept in a student response) while many others were checks for references to keywords or variations on keywords (for example, references to overfitting in questions about improving algorithmic performance).

Evaluation of generated summaries measured their relevancy to the question and KC-tagged feedback on student responses used to generate them, their correctness in aggregating the KC-tagged feedback, and whether or not the format of the output reflected instructor preferences. Relevancy of summaries was determined by whether or not a generated summary only contained content directly related to the topic of the question. Correctness was evaluated based on how well the summary qualitatively reflected the question and applied KC tags. Format was evaluated based on criteria identified by two instructor interviews. All evaluations were given a final Boolean score of TRUE if the criteria for the evaluation were met or FALSE if they did not meet the criteria.

Results

Processing question and student data resulted in 27 topic-level sets of KCs (one for each question), 1,322 KC-tagged evaluations of student responses, and 27 topic-level summaries of student performance (one for each question). 20% of each dataset were used during the first round of establishing IRR.

Knowledge Component Generation

Establishment of IRR on the generated KC dataset (6 generated KC sets) was successful in the first round on the rele-

Question	Count of Student Responses
How [do you] calculate the misclassification error at node t?	50
What is the main difference between these four data types (Nominal, Ordinal, Interval, Ratio)? Please describe them in turn and compare them to the data types that preceded each category.	50
What is [the] Curse of Dimensionality?	50
What is underfitting and overfitting? What usually causes them to occur?	50
What is the typical structure of a decision tree?	50
Why discretize?	50
Why [use] Dimensionality Reduction?	50
What are the main difference between Misclassification Error, Gini Index, and Entropy & Information Gain?	49
What are the advantages of hierarchical clustering over partitional clustering?	49
How do soft-margin SVMs differ from hard-margin SVMs?	49
What are the steps of *k*-fold cross-validation?	49
What scenarios should we focus on False Negative during model evaluation? Please provide at least 2 examples.	49
What is the purpose of the margin in a SVM?	49
What is the primary difference between partitional and hierarchical clustering?	49
What is an example of an association rule in a market-basket analysis, and how can it be useful?	49
What are Eigenvectors?	49
What are the Common Properties of Similarity?	49
What effect does the size of the value of K in the KNN algorithm usually have on its results?	49
What are the steps to construct an ROC curve?	49
How does regularization improve linear regression models?	48
How does logistic regression differ from linear regression, and when should it be used?	48
How does the alpha value reflect the importance of a weak classifier in AdaBoost?	48
How does gradient descent help train artificial neural networks?	48
What is the primary purpose of linear regression, and when is it most useful?	48
What is the main advantage of adding hidden layers to an artificial neural network?	48
What is the primary purpose of using convolutional layers in deep learning?	48
Why does Bagging improve classifier performance?	48

Figure 5: Count of responses to questions posed to an undergraduate class using the ExplainIt tool.

vancy of generated KCs to their question and the sufficiency of generated KCs in capturing whether or not a student created a successful response to a question. IRR was exceptionally high for these two criteria (AC1=1 for both). Scope required further discussion and an additional round of coding on 10% of the dataset to resolve discrepancies, at which point IRR was sufficiently reached (AC1=0.7561).

The LLM generated from 7 to 17 KCs per question, with a median of 10 KCs produced. Only one generated KC set contained substantial topically-irrelevant content ("What are the Common Properties of Similarity?"). All generated KCs were found to be sufficient to identify correct responses. All generated KCs contained content that was out of the scope of the question to some degree. The percentage of out-of-scope KCs ranged from 9% to 90% with a median of 30%.

Knowledge Component Tagging

The tagging accuracy of KCs ranged from 0.10 to 1.00, with a median accuracy of 0.88. Precision ranged from 0.00 to

1	Definition of *k*-fold cross-validation*
2	Purpose of *k*-fold cross-validation*
3	Selecting the number of folds (k)
4	Splitting the dataset into k subsets
5	Training and validation process
6	Iterative process of model evaluation
7	Averaging performance metrics
8	Handling data imbalance (if relevant)*
9	Importance of random selection in folds
10	Comparison with other validation techniques (if relevant)*

Figure 6: Example generated KCs for the question "What are the steps of *k*-fold cross-validation?" Out-of-scope KCs are marked with an asterisk. Although one could argue that a definition of *k*-fold is implicit in outlining the steps of *k*-fold, this KC was not deemed explicitly relevant in this case as a definition may cover content beyond steps.

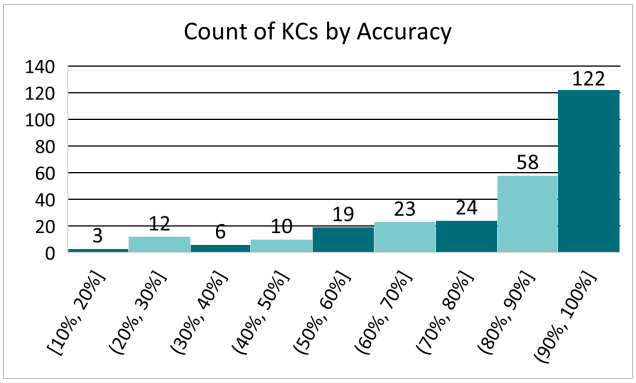

Figure 7: Distribution of tagging accuracies among knowledge components.

1.00 with a median of 0.97 across all KCs. Recall again ranged from 0.00 to 1.00 but with a median of 0.89. No discrepancies were observed with regards to the LLM hallucinating additional or fewer KCs than were provided as input. A histogram of tagging accuracies among KCs is shown in Figure 7. 135 KCs had an accuracy of 0.90 or higher.

Summarization and Formatting for Instructor Use

Establishment of IRR on the generated summaries dataset was successful in the first round on the relevancy of generated KCs to their question, the sufficiency of KCs in evaluating whether or not a student created a successful response to a question, and the format of the generated summaries (AC1=1, AC1=1, and AC1=0.7561, respectively). An example summary is shown in Figure 8. All summaries were found to be relevant and correct with regards to their respective instructor question and matching KC-tagged feedback on student responses. No summaries were found to have omitted important content, such as any KCs that students performed poorly on, although the LLM occasionally merged multiple KCs into a single bullet point or sentence when reporting. Summaries were found to be overly critical in their wording and did not emphasize student successes — only deficiencies. Inconsistencies were also noted in the application of specific wordings to the number of students that did not include a given KC. For example, the LLM may sometimes use the wording "...many students..." to indicate anywhere from 40% to 100% of students, or "...several students..." to indicate 92% of students.

Some minor errors were found in formatting (2 affected summaries), but these were due to output token restrictions. Interviews with two instructors identified a desirable set of features for generated summaries: (1) a short overall description of student performance, (2) a list-style format expanding on the short overall description pointing out any notable features of students' performance in a few sentences per list entry, and (3) the ability to drill down into individual student responses to inspect errors that they have made (or that the LLM claims they have made). All outputs reflected features (1) and (2) above despite not specifically prompting the LLM for such output. Feature (3) above was met by default,

> The feedback indicates that students generally understand the definitions of Misclassification Error, Gini Index, Entropy, and Information Gain, along with some basic relationships between these metrics. However, many students struggle with:
>
> 1. **Calculations**: Most responses lack specific calculations for the metrics, which are essential for demonstrating a comprehensive understanding of these concepts.
>
> 2. **Comparative Analysis**: Few students effectively compared Misclassification Error, Gini Index, and Entropy, which is crucial for understanding their relative strengths and weaknesses.
>
> 3. **Applications and Pros/Cons**: Most responses did not discuss the practical applications of these metrics in decision tree algorithms or evaluate their advantages and disadvantages, which limited the depth of their understanding.
>
> Overall, students demonstrated partial comprehension but missed key components that would enhance their understanding of classification metrics and their implications in machine learning.

Figure 8: An abbreviated example of a summary generated from assessments of student responses.

as our tagging system would allow for the ExplainIt system to produce a dashboard of responses by KC tag (or sets of KC tags).

Discussion

This study yielded several insights into the performance of the novel summarization pipeline developed for the tasks of generating KCs relevant to a question, tagging student responses with these KCs, and then summarizing student performance with the ExplainIt CRS. Overall, the pipeline performed well. The KC generation, KC tagging, and summarization steps all largely functioned with few major issues and high relevancy, correctness, and usefulness of output, where applicable.

Generated knowledge components were universally deemed by expert graders and reference to course materials as sufficient to identify a correct student response to a given question by their presence. However, the proposed KCs always exceeded the scope of the question in varying undesirable ways. These out-of-scope KCs likely represent model biases in output format. For example, every KC set contained one or more similarly-worded variations on a definition of the topic(s) being asked about in a question, even if the question is not explicitly (or implicitly) asking for students to produce a definition of a term or terms.

KC tagging reached a high median accuracy of 0.88, median precision of 0.97, and median recall of 0.89 indicating a high level of reliability for most KCs. Some patterns in

KCs with below-median accuracy were observed. "Definition of concept" tags frequently disagreed with the human grader, potentially due to the short, informal nature of the definitions given. Many errors in tagging also appear to be due to whether a KC was implicit in a student's explanation vs. explicit. For example, in questions that asked students to compare and contrast between two algorithmic approaches, a student could implicitly demonstrate understanding of the disadvantages of one algorithm by only outlining the advantages of another. While the student never fully explains the advantages of one of the algorithms over the other in this case, their answer is still effectively correct in this context since an advantage in one algorithm implies a disadvantage in another and students responding to this type of system are under time pressure during a lecture.

The summarization step performed quite well, with no major hallucinations observed and formatting criteria desired by the interviewed teachers being met. As noted in the results, the summaries produced by this system meets by design instructor needs with regards to being able to inspect student responses in detail due to the KC tagging system. Minor token limit errors in format could be solved easily by increasing the maximum length of allowed output; while this could increase usage costs, the real-time processing cost of all KCs, taggings, and summaries was approximately $0.16 USD, so minor increases in length would likely result in negligible cost increases.

Limitations

While the class analyzed in this study produced a wealth of data, the size of generated KC sets and summaries is relatively small (27 artifacts each). This could be improved in at least two ways; firstly, by expanding the dataset, or secondly, by producing several pieces of output per input. The latter would have the added bonus of allowing for the evaluation of consistency of output, which is critical when using probabilistic tools such as LLMs to produce output. Collection and labeling of data is to be concluded this year, opening up a larger set of data spanning multiple disciplines for further exploration of the transferability of this system to other disciplines or contexts.

Built-in model biases were likely present in the output. Words and phrasings like "ideal response" put any output at the mercy of the model's training as to what it interprets as "ideal", which is risky given that LLMs are known to rate their own output more highly than other text of equal quality (Panickssery, Bowman, and Feng 2024). While these biases can be useful in creating output with a consistent format (as in the case of the topic-level summaries), this can lead to undesirable behaviors in KC extraction. While the generated components were largely related to the topic at hand in this study, they frequently went beyond the spirit of the question (such as the consistent demand for definitions on questions that did not necessarily require a definition to demonstrate student understanding of the concept at play).

Lastly, the use of the words "knowledge components" is fraught in this context. Without further evaluation of generated KCs using knowledge tracing or student modeling, it is difficult to identify "true" KCs that are useful reflections of student knowledge gain. This type of analysis has not been performed at this time, but is likely feasible if student response evaluations are expanded to include human ratings of response correctness.

Conclusions and Future Work

Classroom response systems that incorporate self-explanation questions can foster constructive, interactive, and active learning by prompting students to articulate their understanding in their own words. Leveraging LLMs enables real-time, automated assessment, feedback generation, and instructor support, making it feasible to analyze and summarize student responses at scale. However, the use of probabilistic systems in assessment must be done with care in order to minimize potential harm and ensure fairness and transparency. Our work introduced a novel self-explanation evaluation pipeline that integrates summarization to support instructors, encompassing three key tasks: generating KCs relevant to a question, tagging student responses with these KCs, and summarizing student performance with the ExplainIt CRS. Evaluation results indicate that the LLM-generated KCs were largely accurate, relevant, and aligned with instructional goals, enabling effective assessment of student responses. KC-level evaluations showed high agreement with human judgments, indicating that this method is promising. The aggregated summaries successfully highlighted class-wide misconceptions for real-time teaching support.

There are many avenues for the improvement of this CRS. A trained or more specialized out-of-the box model or RAG system using course materials may be able to address the generation of out-of-scope KCs. The validity of generated KCs in this novel in-class context should be addressed as well by statistically evaluating generated KC effectiveness at predicting student performance. The overly-critical tone of the summaries may require additional prompting refinements or the utilization of a fine-tuned model such that student successes are also noted. Furthermore, there is likely a need to enforce consistency of word choices in the summaries. This could involve restricting the use of phrases like "...many students..." to a certain percent range of student errors by breaking the summary creation step into sub-steps where each KC is processed by an algorithm that computes KC correctness and then uses pre-defined wordings to control LLM output. Lastly, a CRS like ExplainIt could be expanded into a longitudinal system to trace student performance over the duration of a course to yield richer insights into student learning.

Acknowledgments

This research was supported by funding from the National Science Foundation under grants DUE-2111473 and DUE-2111216. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Beauchamp, G.; and Kennewell, S. 2010. Interactivity in the classroom and its impact on learning. *Computers Education*, 54(3): 759–766. Learning in Digital Worlds: Selected Contributions from the CAL 09 Conference.

Bewersdorff, A.; Seßler, K.; Baur, A.; Kasneci, E.; and Nerdel, C. 2023. Assessing student errors in experimentation using artificial intelligence and large language models: A comparative study with human raters. *Computers and Education: Artificial Intelligence*, 5: 100177.

Blasco-Arcas, L.; Buil, I.; Hernández-Ortega, B.; and Sese, F. J. 2013. Using clickers in class. The role of interactivity, active collaborative learning and engagement in learning performance. *Computers Education*, 62: 102–110.

Chi, M. T.; and Wylie, R. 2014. The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. *Educational Psychologist*, 49(4): 219–243.

Choi, S. P. M.; Lam, S. S.; Li, K. C.; and Wong, B. T. M. 2018. Learning analytics at low cost: At-risk student prediction with clicker data and systematic proactive interventions. *Journal of Educational Technology Society*, 273–290.

Das, S.; and Lim, L. H. I. 2024. Enhancing Civil Engineering Education: A Comprehensive Analysis of Student Perspectives on Technology-Integrated Learning. In *2024 IEEE Global Engineering Education Conference (EDUCON)*, 1–4.

Dong, J.-J.; Hwang, W.-Y.; Shadiev, R.; and Chen, G.-Y. 2017. Pausing the classroom lecture: The use of clickers to facilitate student engagement. *Active Learning in Higher Education*, 18(2): 157–172.

Duan, Z.; Fernandez, N.; Narayanan, A. B. L.; Hassany, M.; de Alencar, R. S.; Brusilovsky, P.; Akram, B.; and Lan, A. 2025. Automated Knowledge Component Generation and Knowledge Tracing for Coding Problems. *arXiv*.

Dbiec, P. 2018. Effective Learner-Centered Approach for Teaching an Introductory Digital Systems Course. *IEEE Transactions on Education*, 61(1): 38–45.

England, B. J.; Brigati, J. R.; and Schussler, E. E. 2017. Student anxiety in introductory biology classrooms: Perceptions about active learning and persistence in the major. *PLOS One*, 12(8).

Garces, S.; Ravai, G.; Vieira, C.; and Magana, A. J. 2019. Effects of Self-explanations as Scaffolding Tool for Learning Computer Programming. In *2019 IEEE Frontiers in Education Conference (FIE)*, 1–6.

Grissom, S.; Mccauley, R.; and Murphy, L. 2017. How Student Centered is the Computer Science Classroom? A Survey of College Faculty. *ACM Trans. Comput. Educ.*, 18(1).

Gwet, K. L. 2002. Inter-Rater Reliability: Dependency on Trait Prevalence and Marginal Homogeneity.

Hubbard, J. K.; and Couch, B. A. 2018. The positive effect of in-class clicker questions on later exams depends on initial student performance level but not question format. *Computers Education*, 120: 1–12.

Hung, H.-T. 2016. Clickers in the flipped classroom: bring your own device (BYOD) to promote student learning. *Interactive Learning Environments*.

Hunsu, N. J.; Adesope, O.; and Bayly, D. J. 2016. A meta-analysis of the effects of audience response systems (clicker-based technologies) on cognition and affect. *Computers Education*, 94: 102–119.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.*, 55(12).

Kerslake, C.; Denny, P.; Smith, D. H.; Leinonen, J.; MacNeil, S.; Luxton-Reilly, A.; and Becker, B. A. 2025. Exploring Student Reactions to LLM-Generated Feedback on Explain in Plain English Problems. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, SIGCSETS 2025, 575–581. New York, NY, USA: Association for Computing Machinery. ISBN 9798400705311.

Kirstein, F.; Wahle, J. P.; Gipp, B.; and Ruas, T. 2025. CADS: A Systematic Literature Review on the Challenges of Abstractive Dialogue Summarization. *J. Artif. Int. Res.*, 82.

Lim, W. N. 2017. Improving student engagement in higher education through mobile-based interactive teaching model using socrative. In *2017 IEEE Global Engineering Education Conference (EDUCON)*, 404–412.

Liu, C.; Chen, S.; Chi, C.; Chien, K.-P.; Liu, Y.; and Chou, T.-L. 2017. The Effects of Clickers With Different Teaching Strategies. *Journal of Educational Computing Research*, 55(5): 603–628.

Milo D. Koretsky, B. J. B.; and Higgins, A. Z. 2016. Written justifications to multiple-choice concept questions during active learning in class. *International Journal of Science Education*, 38(11): 1747–1765.

Moon, H.; Davis, R. L.; Neshaei, S. P.; and Dillenbourg, P. 2025. Using Large Multimodal Models to Extract Knowledge Components for Knowledge Tracing from Multimedia Question Information. In Mills, C.; Alexandron, G.; Taibi, D.; Bosco, G. L.; and Paquette, L., eds., *Proceedings of the 18th International Conference on Educational Data Mining*, 342–353. Palermo, Italy: International Educational Data Mining Society. ISBN 978-1-7336736-6-2.

Moore, S.; Schmucker, R.; Mitchell, T.; and Stamper, J. 2024. Automated Generation and Tagging of Knowledge Components from Multiple-Choice Questions. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, L@S '24, 122–133. New York, NY, USA: Association for Computing Machinery. ISBN 9798400706332.

Oliveira Moraes, L.; and Pedreira, C. E. 2021. Clustering Introductory Computer Science Exercises Using Topic Modeling Methods. *IEEE Transactions on Learning Technologies*, 14(1): 42–54.

OpenAI. 2023. GPT-4 Technical Report. https://cdn.openai.com/papers/gpt-4.pdf. Accessed: 2025-04-23.

OpenAI. 2025. ChatGPT-4o-mini API. https://platform.openai.com/docs. Accessed: 2025-04-23.

Panickssery, A.; Bowman, S. R.; and Feng, S. 2024. LLM Evaluators Recognize and Favor Their Own Generations. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 68772–68802. Curran Associates, Inc.

Papadopoulos, P. M.; Natsis, A.; Obwegeser, N.; and Weinberger, A. 2018. Enriching feedback in audience response systems: Analysis and implications of objective and subjective metrics on students' performance and attitudes. *Journal of Computer Assisted Learning*, 305–316.

Petrović, J.; and Pale, P. 2021. Achieving Scalability and Interactivity in a Communication Skills Course for Undergraduate Engineering Students. *IEEE Transactions on Education*, 64(4): 413–422.

Pisheh, E. A. G.; NejatyJahromy, Y.; Gargari, R. B.; Hashemi, T.; and Fathi-Azar, E. 2018. Effectiveness of clicker-assisted teaching in improving the critical thinking of adolescent learners. *Journal of Computer Assisted Learning*, 82–88.

Renaud, K.; and Cutts, Q. 2013. Teaching human-centered security using nontraditional techniques. *ACM Trans. Comput. Educ.*, 13(3).

Rodríguez-Martínez, J. A.; González-Calero, J. A.; del Olmo-Muñoz, J.; Arnau, D.; and Tirado-Olivares, S. 2022. Building personalised homework from a learning analytics based formative assessment: Effect on fifth-grade students' understanding of fractions. *British Journal of Educational Technology*, 76–97.

Shi, Y.; Schmucker, R.; Chi, M.; Barnes, T.; and Price, T. 2023. KC-Finder: Automated Knowledge Component Discovery for Programming Problems. In Feng, M.; KÃ¤ser, T.; and Talukdar, P., eds., *Proceedings of the 16th International Conference on Educational Data Mining*, 28–39. Bengaluru, India: International Educational Data Mining Society. ISBN 978-1-7336736-4-8.

Stains, M.; Harshman, J.; Barker, M. K.; Chasteen, S. V.; Cole, R.; DeChenne-Peters, S. E.; Eagan, M. K.; Esson, J. M.; Knight, J. K.; Laski, F. A.; Levis-Fitzgerald, M.; Lee, C. J.; Lo, S. M.; McDonnell, L. M.; McKay, T. A.; Michelotti, N.; Musgrove, A.; Palmer, M. S.; Plank, K. M.; Rodela, T. M.; Sanders, E. R.; Schimpf, N. G.; Schulte, P. M.; Smith, M. K.; Stetzer, M.; Valkenburgh, B. V.; Vinson, E.; Weir, L. K.; Wendel, P. J.; Wheeler, L. B.; and Young, A. M. 2018. Anatomy of STEM teaching in North American universities. *Science*, 359(6383): 1468–1470.

Touvron, H.; Martin, L.; Stone, K. R.; Albert, P.; Almahairi, A.; Babaei, Y.; lay Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D. M.; Blecher, L.; tian Cantón Ferrer, C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A. S.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I. M.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M. H. M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *ArXiv*, abs/2307.09288.

Wang, A. Y.; Chen, Y.; Chung, J. J. Y.; Brooks, C.; and Oney, S. 2021. PuzzleMe: Leveraging Peer Assessment for In-Class Programming Exercises. *Proc. ACM Hum.-Comput. Interact.*, 5(CSCW2).

Wei, Y.; Carvalho, P.; and Stamper, J. 2025. KCluster: An LLM-based Clustering Approach to Knowledge Component Discovery. In Mills, C.; Alexandron, G.; Taibi, D.; Bosco, G. L.; and Paquette, L., eds., *Proceedings of the 18th International Conference on Educational Data Mining*, 228–240. Palermo, Italy: International Educational Data Mining Society. ISBN 978-1-7336736-6-2.

Williams, B.; Lewis, B.; Boyle, M.; and Brown, T. 2011. The impact of wireless keypads in an interprofessional education context with health science students. *British Journal of Educational Technology*, 337–350.