2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #51443

AUTOMATED GENERATION OF MIXED-TYPE QUESTIONS FOR ONLINE QUESTION POOLS

Dr. John T. Tester, Tennessee Technological University

Dr. Tester has expertise in Engineering Design with interests in additive manufacturing, traditional manufacturing processes, engineering management, biomechanics, and engineering education. He also has been a faculty advisor for student teams engaged in SAE intercollegiate competitions, such as Baja and Aero Design, for over 25 years.

Automated Generation of Mixed-Type Questions for Online Question Pools

Abstract

Online courses for engineering have a challenge in that the conceptual learning is matched by often analytical or mathematical problem solving. Given the current availability of online solution databases to students, and the difficulty of proving any student is not using such databases in solving assignments or exam questions, an instructor must often construct questions that are unique to individual students, even when they are in the same online enrolled class. Presented is a method of rapidly generating alternatives of the same question for online question deployment, using readily available office software. This approach, once applied to a complex question creation the first time, can generate many multiples, even hundreds, of similar questions with varied parameters, each requiring unique answers. Targeted for a Learning Management System (LMS) using the "BrightSpace®" or D2L environment, this technique uses Microsoft Word Mail Merge and Respondus. The use of Microsoft Excel spreadsheet is also inserted for mathematically-oriented questions or sub-questions in a D2L quiz. Modifications of the basic technique allows simple HTML blocks to be duplicated within these questions for display in the D2L question and answer blocks as needed. The use of the spreadsheet allows for complex questions to be mathematically solved off-line, recorded for 'regular expression' checking, varied for multiple questions in a pool, and then processed into D2L. Through the use of 'fill-in-the-blank' type D2L quiz questions, the instructor is allowed to offer multiple types of answers within a single problem statement, the types being textual responses, combined with separate mathematical responses. This last characteristic is impossible for a D2L mathematically-based type of question, but can be emulated as being possible with this question-construction technique. The use of this technique is appropriate for large online enrollments, though it is acknowledged that the initial set up can be potentially time-consuming.

Keywords

Online, asynchronous, engineering education, engineering management, D2L

Introduction

All higher-institution educational organizations were significantly impacted by the relatively recent COVID-19 restrictions in 2019-2020, resulting in a shift to online education methods. Though the shift may be temporary, increased observations of cheating was occurring even before the pandemic [1]. One method of combating the cheating approach of sharing online homework and exam questions is by creating pools of different questions on same topic, randomly assigned to students, for assignments and exam [2]. This method also dissipates the uploading of solutions by students to solution-posting websites. But this method results in an increased workload for the instructor in terms of generating the separate questions, programming or inputting the separate questions in their learning management system (LMS), and afterwards, grading the different questions with separate rubrics. A way to mitigate these issues is by generating the variations in the questions in an automated or semi-automated

manner, and using the LMS system to conduct automatic grading. As a bonus, early online assignment advocates found improved student academic performance with online assignments [3].

The method introduced here was first attempted in Spring of 2022, for several engineering management and project management graduate courses in the newly started Master's of Science in Engineering Management (MSEM). The MSEM program was launched as a completely online and asynchronous program of study. The primary goal was to provide an early-career, technology-oriented degree to professionals that address leadership, management, and technical knowledge. The author, an instructor for several classes, was informed that potentially large numbers of online students would enroll within three years after the 2020 program start. Given there was no on-campus Engineering Management or Industrial Engineering program from which to recruit campus student graders or assistants, the instructor developed this question-generation method as a pre-emptive approach to manage the anticipated increasing grading workload.

The presented method is a version of an "Automated Grading System" (AGS), where questions are generated from a template, so that grading can be automated from the template as well [12]. The questions, with a grading template and student feedback, are then uploaded into the LMS for online use. This approach was developed without the assistance of "Artificial Intelligence" (AI) mainly because AI was not integrated into the university's D2L LMS at the start of the AGS development process for the online degree program. The use of AI in engineering education applications is a topic all on its own; that area of research is particularly intensive in allowing educators to both generate and grade word-answer or essay-style questions due to its language interpretation abilities. [12, 13, 14]. The types of questions for the subject online classes have "…clear-cut correctness criteria or fixed solutions…" thus encouraging a non-AI, template-based AGS approach [15]. This philosophy has been used prior to the advent of AI LMS integration with success in programming code grading as well.

Background

The LMS at the subject institution is the D2L Brightspace® system, termed "iLearn" at that university. It is used for all on-campus and online classes in the delivery of electronic-based communications. Specific to this topic: D2L contains a quiz tool that allows the instructor to create different types of questions to be inserted into online homework assignments and exams. ILearn can accommodate True/False, Multiple Choice (MC), Written Response (WR), Short Answer, Fill in the Blanks (FIB) Arithmetic (2+2) and Likert questions, among others [4]. With the exception of the WR questions (where an essay style answer is required), the iLearn quiz tool automatically grades the student submissions to the questions. How well this grading is accomplished is based largely upon the instructor's skill in anticipating all possible student answers, right and wrong. Deterministically oriented questions, where only a single narrow answer is correct, are always graded correctly once programmed. Such question types include True/False and MC. Unfortunately, these two types of questions also tend to be limited in stimulating student thinking when they answer them. Arithmetic questions also have discrete answers possible

when set up, though the instructor must take care that the students clearly recognize the need to answer precisely to the correct decimal and or units.

The SA and FIB questions in particular may be programmed to match words or phrases entered by the student, and have points automatically awarded if correct. This programming subtext is termed *regular expression (regex)* pattern matching. Regex was prominent as a script programming pattern matching approach (in layman's terms) prior to online automated grading adaptation. It has since been heavily examined and adapted for Massive Open Online Courses (MOOC) [5]. Though not a panacea for all online questioning, it does have promise in guiding automated grading for SA and FIB types of questions.

Coming back to the iLearn context of question construction, there is a need for multiple types of required answers to be in a single question statement. In courses requiring analytically-oriented programs of study, often questions require a combination of math and text answers. For example, a quiz or example question may require a mathematically-created answer, followed by a short, word-oriented sentence answering why that math answer should be correct. Frequently in engineering management, one answer is based upon the outcome of the other, so the question must have two types of answers to be asked: A math question, requiring a mathematical solution, and a choice to be made. One common and specific example is that of comparing the economic value of several projects. A student may be required to calculate the present value of three projects, and then provide a choice as to the best project to select. They may even be required to type in a brief statement as to why they make that selection. This example would thus require three math (2+2) questions, one FIB (or SA) question, and then a final FIB or SA question. However, D2L can only accommodate one answer per math question [6]. The three math answers are not possible, nor are they allowed in combination with required text answers. Yet, such a question must be created in order to allow for a combination of analytical and decision-making skills to be displayed in a single problem.

Compounding the above issue is the desire to have variations of mixed-answer-type questions, so that an instructor can have the same basic question asked of different students simultaneously, yet each not have exactly the same required answers. This approach, as noted earlier, enhances academic security for online questioning within a class.

It seems that D2L does not readily issue information on how to accommodate these and other problems to its users. An alternative LMS is Moodle, which requires some knowledge of script programming [7]. However, this source's author asserts that all university faculty should not have programming skills in order to develop course delivery in their expertise. D2L seems to be developed with that idea in mind, having graphical interfaces for the structured question types noted earlier.

Educators have developed work-arounds over the years, attempting to have the complexity of mixed-type questions, yet only needing limited programming skills to create them. One educator in particular,

Amy Goldlist, caught this author's attention through her blog posts that addressed creating multiple questions with the coordinated use of three software programs that are mostly familiar to many higher-institution educators: Microsoft (MS) Word, MS Excel, and Respondus® [8]. The first two programs are widely used in the academic domain as a word processor and spreadsheet, respectively. The last software has become known to many online educators for a variety of functions, but the function of value to this subject is as a preprocessor of Word documents into LMS packages for uploading to the system. This approach was where the author began to develop a method for the Engineering Management courses needed in the new program.

Approach

The original, posted, method for constructing these multi-type questions is to use the FIB question exclusively; the numerical or textual answers required are all graded via text-based string checking. This author enhanced this approach by using regex programming for more flexibility in the answers for the multi-part questions. Also added was a method to insert HTML-based feedback and image insertions to the question text. The primary issue is to generate multiple questions that have the same basic format, but different variables and thus possible answers. An overview of this process is illustrated in a flowchart in Figure 1.

Creating the question

The question is created by the instructor, of course. The instructor will identify which numbers and words will change with each variation of the question; i.e., determine the variables, whether they are numerical or textual. As noted earlier, the iLearn quiz tool question type is FIB for generating multiple answers with (apparently) mixed-type questions. An example question will be the basis for the method illustration.

> *Your company asks you to conduct a Net Present Worth analysis on a potential product development opportunity. The investment is **$1,000,000** for the product at the start of this year. The marketing department assures the company that the product will generate **$200,000** profit at the end of each year for **8** years. The company requires a **15%** return on investment (ROI) in order to accept the project. Ignore any tax considerations.*
>
> *For full credit to this problem, upload (or type into HTML box if appropriate) your supporting calculations and reasoning to the previously displayed question.*
>
> *a. What is the NPW of the project?*
>
> ***$897,464***
>
> *b. Should your company invest in the project, yes or no?*
>
> ***NO***
>
> *General Feedback:*

Using the ROI as the interest rate i, compounded annually, and abbreviating long numbers initially,

a.

*NPW = -**$1M** + **$0.200M**/(1+**0.15**)^1+ **$0.200M**/(1+**0.15**)^2+ **$0.200M**/(1+**0.15**)^3+ **$0.200M**/(1+**0.15**)^4+ **$0.200M**/(1+**0.15**)^5+ **$0.200M**/(1+**0.15**)^6+ **$0.200M**/(1+**0.15**)^7+ **$0.200M**/(1+**0.15**)^8*

*NPW = **$1,000,000** - **$897,464** = **-$102,536***

b. ***NO***

Since NPW < 0, Choose NOT to invest in project

The variables in the question have been identified with bold font; the entire last line for *b.* is a text 'variable' in that the line is given dependent upon the answer of either NO or YES, previously.

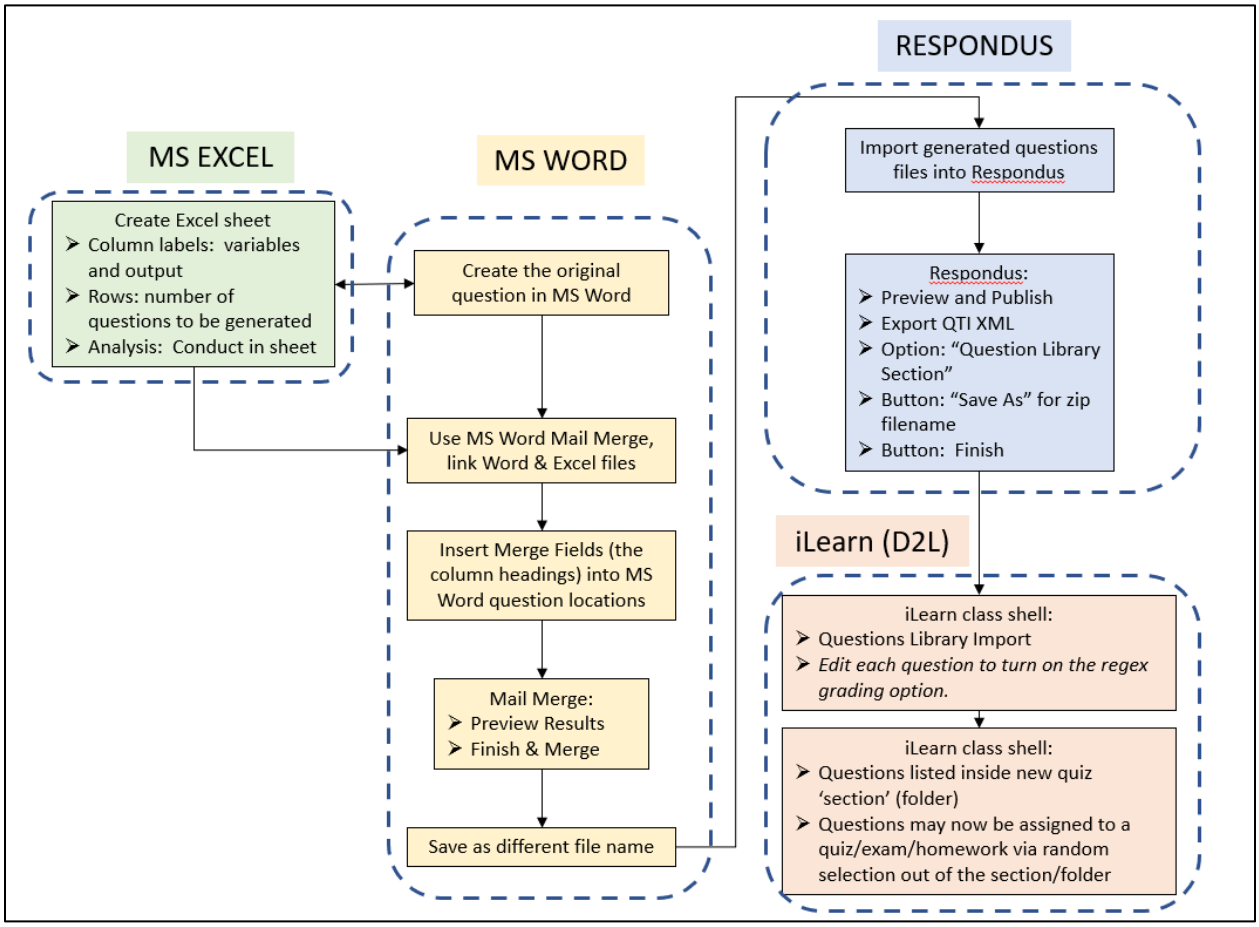


Figure 1. Flowchart of method; colors with corresponding titles indicate the four software packages used.

"General Feedback" in a question is typically used by the instructor to provide general information to the entire student population, and a separate "Answer" block is filled in the quiz tool for the answer to the question, provided to students after all quizzes are graded. However, the instructor uses the General Feedback block itself as a means by which to provide an answer which is custom to each varied instance of this particular question. Thus, variables will be provided in the General Feedback section of the iLearn question as well. There may be more elegant formatting available, but this simple question will serve as the example.

Microsoft (MS) Word is used as the common software tool for this exercise. It has an important function which was originally created and still used in physical mailings: "Mail Merge". The Mail Merge feature was created to set up mailings using merge fields for addresses, names, and other identifying, varying data sets, linking through MS Excel. [9]. In the question generation method here, MS Word Mail Merge is used to organize the mathematics in MS Excel to create numerical answers, and MS Excel logic statements to select the word answers. Variables that were intended as address information become variables for values transferred from an MS Excel spreadsheet.

The next step is to replace the hard-coded bold variable values in the example above with variable names. The example below shows the result; the names are coded in a manner that makes logical sense to the author. The reader may find more useful encoding methods for setting up variable names, but the point is to map these variable names to the spreadsheet so that these two are easily linked in a later step. The term "_TEXT" is added by the author in some variable names to enable formatting of numerical answers precisely into desired character text answers, such that they will import properly into the iLearn quiz question later. However, this _TEXT approach is not required to accomplish the question creation.

Type: FMB

Title: NPW1Choice **Qnum**

Points: 4

1. Your company's project allows a software product development, if financially feasible. You are to conduct a Net Present Worth (NPW) analysis on a potential product development opportunity. The investment is **Invest_A_TEXT** for the product at the start of this year. The marketing department assures the company that the product will generate **PayA1_TEXT** profit at the end of each year for **years** years. The company requires a **In_perc_TEXT** return on investment (ROI) in order to accept the project. Ignore any tax considerations.

[HTML]<u>Give numerical answers to second decimal place</u>[/HTML].

For full credit to this problem, upload (or type into HTML box if appropriate) your supporting calculations and reasoning to the previously displayed question in this homework set.

a. What is the NPW of the project?

Answer: [**NPW_A_regexp**]

b. Should your company invest in the project, YES or NO?

Answer: [**Choose_regexp**]

@ General Feedback:

Using the ROI as the interest rate *i,* compounded annually, and abbreviating long numbers initially,

a. For n = years years, . . .

NPW = -**Invest_A_TEXT** + **PayA1_TEXT**/(1+**ln_dec_TEXT**)^1+ **PayA1_TEXT**/(1+**ln_dec_TEXT**)^2+ **PayA1_TEXT** /(1+**ln_dec_TEXT**)^3+ . . . + **PayA1_TEXT** /(1+ **ln_dec_TEXT**)^years

NPW = -**Invest_A_TEXT** + (**PW_RevONLY_TEXT**) = **NPWA_TEXT**

b. «Choose_CAP»

CHOOSE_TEXT_FB

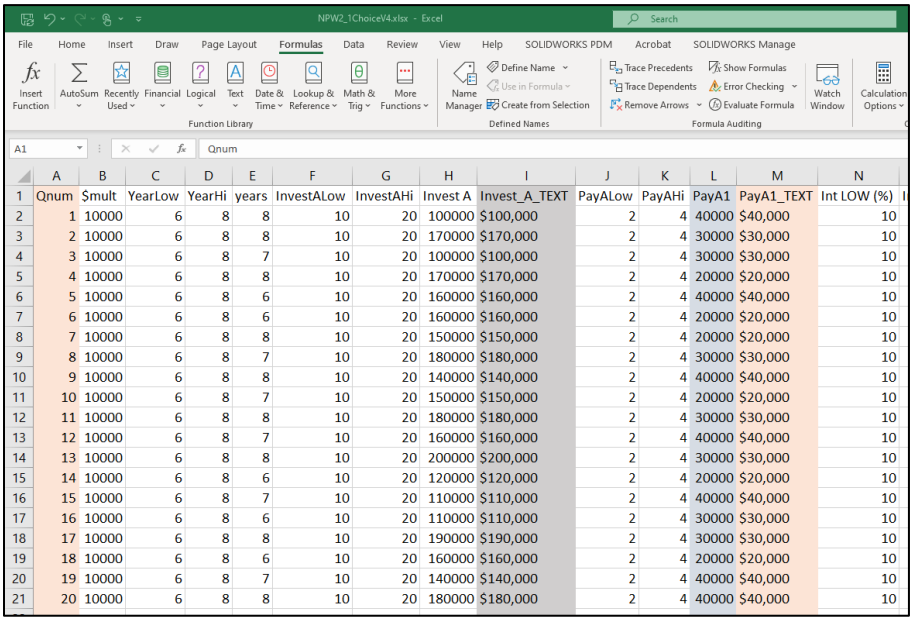
The first three fields are required by a second pre-processing package, Respondus ®, which processes text eventually into questions formatted for insertion into iLearn. FMB represents an FIB question type, the "Title" field identifies the question by a phrase, so that the instructor can identify the question in a database of questions later on. "Qnum" is the question number; when multiples of question variations are generated from the spreadsheet, these numbers are used to distinguish between these variations. The "Points" question represents the number of points assigned to the entire question.

MS Excel is next required by the author in order to organize the question components into the mathematical and textual results. The main concept is that each row in a single tab represents the entire question, with the input variables and output. Column titles will represent the variable names identified in the problem statement. Additional columns are allowed to assist in constructing the final answers, but only the column titles that match those in the question statement will be mapped for the Mail Merge process. In this example, 20 question variations are created. An instructor may hard-code the input variable variations such that pre-determined, yet different outputs, are generated. In this example, however, the MS Excel RAND() function allows for the values to be generated in a range of values. MS Excel also allows for conditional statements to determine what textual answer is required, based upon the numerical result generated from the math portion of the question. Of course, another option is to simply type in each question variation by hand, but that approach would mostly defeat the purpose of the exercise.

Both the auto graded math and textual answers from the students are addressed by using regular expressions. Thus, the format for the resulting answers must be programmed into the MS Excel columns as well. Ultimately, the Mail Merge results in a text output that is then part of an online display to students. Thus, it is recommended that the numerical answers be converted to text answers via the MS Excel TEXT() function. Otherwise, math answers with long decimal results would be inserted into the

webpage with long decimals, whereas for currency answers, frequently only whole dollars are required. Also, the regular expression auto grading needs to be limited to some level of decimal checking, and thus a dollar roundoff can be integrated into the TEXT() formatting.

A portion of the MS Excel file for linking with the MS Word file is shown in Figure 2. For ease of reference, the instructor highlighted those columns that have headers which match the variables (in bold) in the MS Word problem statement. The numbers under these variable names will be input into the 20 questions for this example.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Qnum	$mult	YearLow	YearHi	years	InvestALow	InvestAHi	Invest A	Invest_A_TEXT	PayALow	PayAHi	PayA1	PayA1_TEXT	Int LOW (%)
2	1	10000	6	8	8	10	20	100000	$100,000	2	4	40000	$40,000	10
3	2	10000	6	8	8	10	20	170000	$170,000	2	4	30000	$30,000	10
4	3	10000	6	8	7	10	20	100000	$100,000	2	4	30000	$30,000	10
5	4	10000	6	8	8	10	20	170000	$170,000	2	4	20000	$20,000	10
6	5	10000	6	8	6	10	20	160000	$160,000	2	4	40000	$40,000	10
7	6	10000	6	8	6	10	20	160000	$160,000	2	4	20000	$20,000	10
8	7	10000	6	8	8	10	20	150000	$150,000	2	4	20000	$20,000	10
9	8	10000	6	8	7	10	20	180000	$180,000	2	4	30000	$30,000	10
10	9	10000	6	8	8	10	20	140000	$140,000	2	4	40000	$40,000	10
11	10	10000	6	8	7	10	20	150000	$150,000	2	4	20000	$20,000	10
12	11	10000	6	8	8	10	20	180000	$180,000	2	4	30000	$30,000	10
13	12	10000	6	8	7	10	20	160000	$160,000	2	4	40000	$40,000	10
14	13	10000	6	8	8	10	20	200000	$200,000	2	4	30000	$30,000	10
15	14	10000	6	8	6	10	20	120000	$120,000	2	4	20000	$20,000	10
16	15	10000	6	8	7	10	20	110000	$110,000	2	4	40000	$40,000	10
17	16	10000	6	8	6	10	20	110000	$110,000	2	4	30000	$30,000	10
18	17	10000	6	8	8	10	20	190000	$190,000	2	4	30000	$30,000	10
19	18	10000	6	8	6	10	20	160000	$160,000	2	4	20000	$20,000	10
20	19	10000	6	8	7	10	20	140000	$140,000	2	4	40000	$40,000	10
21	20	10000	6	8	8	10	20	180000	$180,000	2	4	40000	$40,000	10

Figure 2. Portion of MS Excel file to link. Only highlighted values are linked to MS Word file.

Returning to the MS Word document, Mail Merge is now selected to link the MS Word document to the MS Excel file. The MS Excel file is the 'existing list' of recipient from the Mail Merge context, but instead of mailing recipients, one now understands that the list is of variables to select. Once the correct tab is selected and linked, the instructor will select each variable (bold) phrase in the problem statement, execute the "Insert Merge Field" function, then select from the MS Excel header that corresponds to it (see Figure 3). Though it is not required for the MS Excel columns to have the same header text as the variable names typed in the question, it is very helpful for the instructor to follow that procedure, in order to not lose track of which values go to what parts of the original question. For complex analytical questions, this discipline may be a way to avoid losing time due to mapping errors.

Complex word answers may be mapped into the feedback, conditional on the Qnum question row, by generated the entire phrases and sentences into MS Excel, and placing them appropriately under a

Merge Field header column. That approach allows a single variable to represent an entire statement unique to that version of the question. The example in Figure 3 illustrates this context for the very last variable in the problem statement, **CHOOSE_TEXT_FB.**

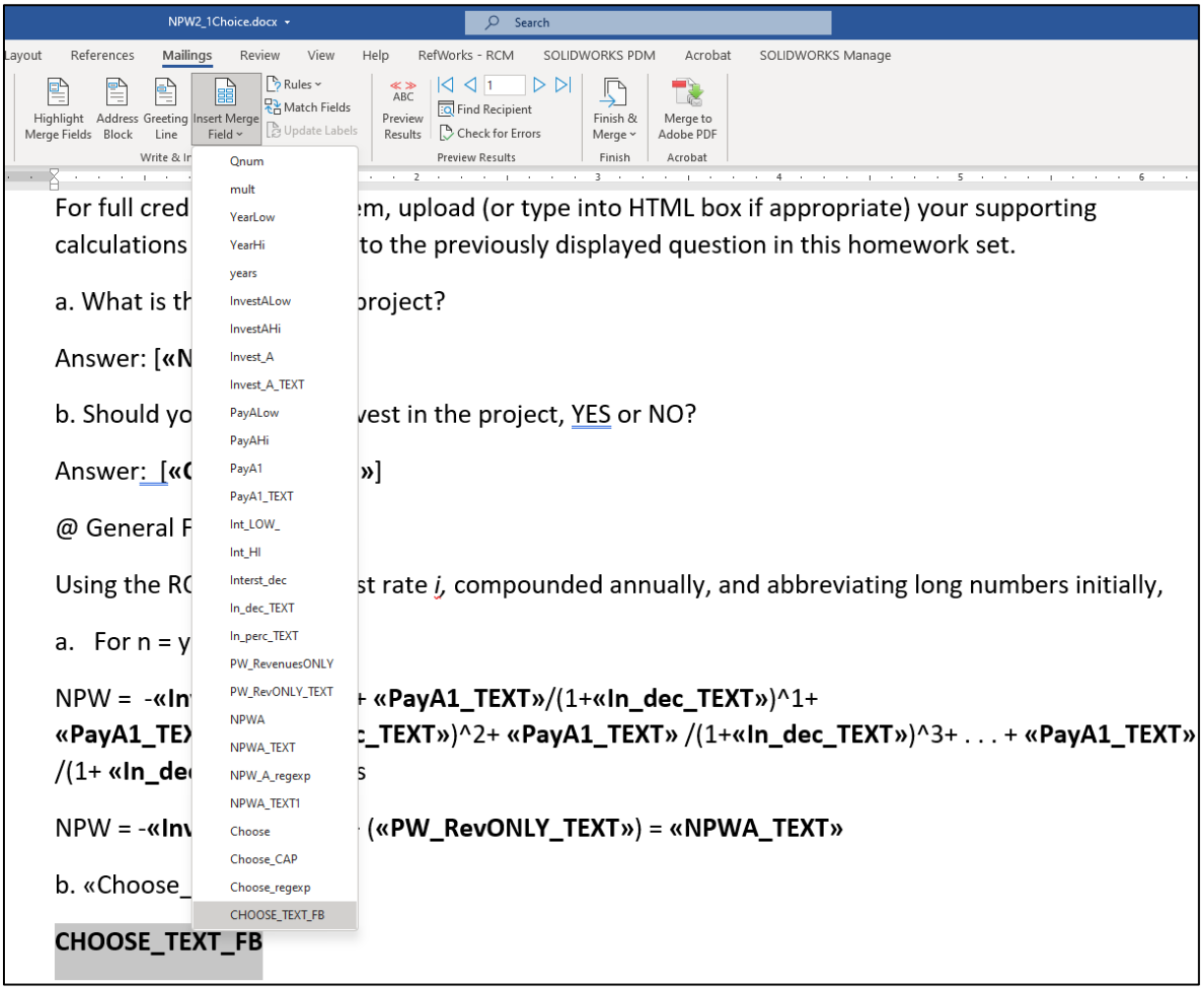
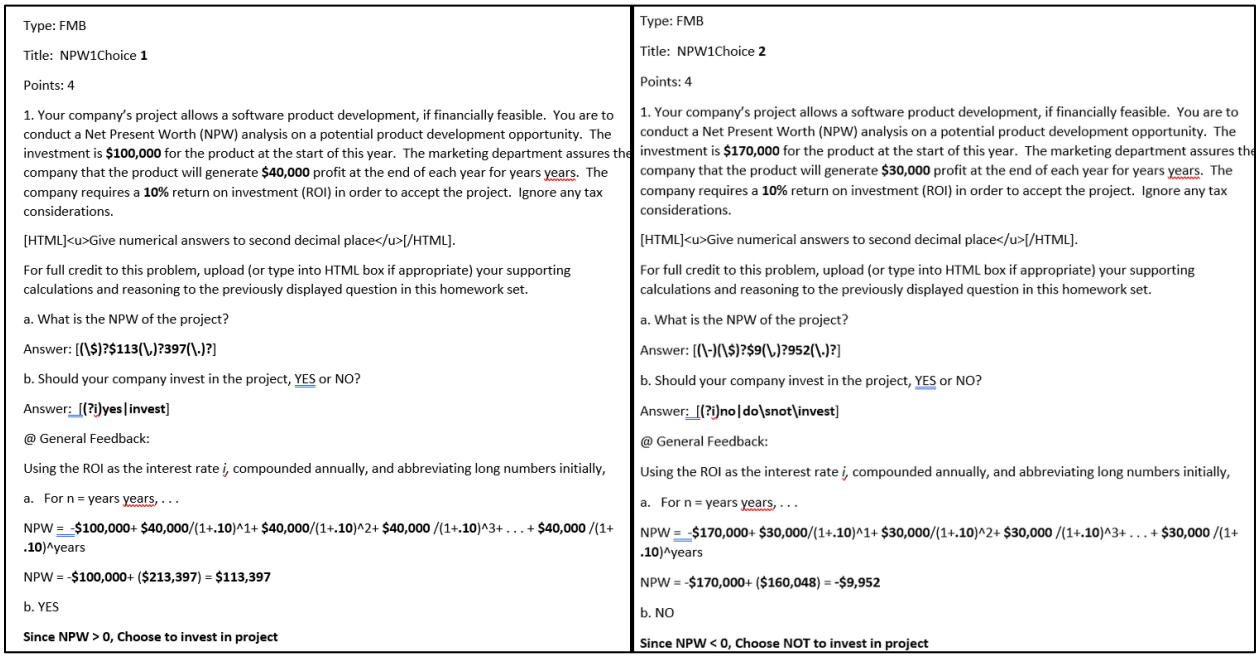


Figure 3. Example of inserting the merge field that is linked to the MS Excel sheet headers (listed in drop down).

With all the problem statement variables linked to the MS Excel values via the column headers, the Mail Merge function is able to generate formatted problem statements. For this example, 20 questions are generated via the Finish & Merge function, allowing for the multiple questions via "Edit Individual Documents." Once accomplished, a new file is generated; each question variation title is modified with an increasing number, from 1 to Qnum. Different questions are generated, with different answers. Questions 1 and 2 are shown for comparison from the current example in Figure 4.

At this point, the text questions' components could be copied and pasted manually into portions of iLearn's quiz tool editor for 20 different questions. However, the goal is to transfer these MS Word-based questions into iLearn in an efficient manner. This transfer formatting is accomplished with the

Respondus software, which is used for several other online LMS platforms for the same purpose. [10], [11].



Type: FMB

Title: NPW1Choice **1**

Points: 4

1. Your company's project allows a software product development, if financially feasible. You are to conduct a Net Present Worth (NPW) analysis on a potential product development opportunity. The investment is **$100,000** for the product at the start of this year. The marketing department assures the company that the product will generate **$40,000** profit at the end of each year for years years. The company requires a **10%** return on investment (ROI) in order to accept the project. Ignore any tax considerations.

[HTML]<u>Give numerical answers to second decimal place</u>[/HTML].

For full credit to this problem, upload (or type into HTML box if appropriate) your supporting calculations and reasoning to the previously displayed question in this homework set.

a. What is the NPW of the project?

Answer: **[(\$)?$113(\.)?397(\.)?]**

b. Should your company invest in the project, YES or NO?

Answer: **[(?i)yes|invest]**

@ General Feedback:

Using the ROI as the interest rate *i*, compounded annually, and abbreviating long numbers initially,

a. For n = years years, . . .

NPW = **-$100,000+ $40,000**/(1+**.10**)^1+ **$40,000**/(1+**.10**)^2+ **$40,000** /(1+**.10**)^3+ . . . + **$40,000** /(1+**.10**)^years

NPW = **-$100,000**+ (**$213,397**) = **$113,397**

b. YES

Since NPW > 0, Choose to invest in project

Type: FMB

Title: NPW1Choice **2**

Points: 4

1. Your company's project allows a software product development, if financially feasible. You are to conduct a Net Present Worth (NPW) analysis on a potential product development opportunity. The investment is **$170,000** for the product at the start of this year. The marketing department assures the company that the product will generate **$30,000** profit at the end of each year for years years. The company requires a **10%** return on investment (ROI) in order to accept the project. Ignore any tax considerations.

[HTML]<u>Give numerical answers to second decimal place</u>[/HTML].

For full credit to this problem, upload (or type into HTML box if appropriate) your supporting calculations and reasoning to the previously displayed question in this homework set.

a. What is the NPW of the project?

Answer: **[(\-)(\$)?$9(\.)?952(\.)?]**

b. Should your company invest in the project, YES or NO?

Answer: **[(?i)no|do\snot\invest]**

@ General Feedback:

Using the ROI as the interest rate *i*, compounded annually, and abbreviating long numbers initially,

a. For n = years years, . . .

NPW = **-$170,000+ $30,000**/(1+**.10**)^1+ **$30,000**/(1+**.10**)^2+ **$30,000** /(1+**.10**)^3+ . . . + **$30,000** /(1+**.10**)^years

NPW = **-$170,000**+ (**$160,048**) = **-$9,952**

b. NO

Since NPW < 0, Choose NOT to invest in project

Figure 4. Questions 1 and 2 generated (out of 20).

The original question was formatted initially in a Respondus-oriented format. For example, the code "@" represents the start of a line that is for feedback. Brackets indicate that an answer is contained. The first three lines at the start indicate the Type of question, the Title, and Points allocated.

Within the Respondus software, the instructor directs it towards importing the MS Word document in a designated format such as .TXT or .DOCX, selecting the Mail Merge generated file. The imported questions are available for review through the Respondus interface, to check for any typographical or format errors from the import process. Then lastly, an XML file is created for export; this file is to be imported into the D2L LMS as a 'question library section.' The XML file is saved intially as a zip file, though with proper Respondus installation, the file could be directly loaded into the D2L LMS from Respondus. This zip file format is more completely stated as a "IMS QTI compliant course package" for this specific iLearn D2L LMS example, where QTI means "Question and Test Interoperability". IMS Global, now 1Edtech, is a non-profit collaborative organization that addresses digital learning environments; it has developed several standards related to digital learning and educator training.

Results and Adjustments

The XML ZIP file imports into iLearn as a Question Library section (a 'folder' to the instructor layman). A section may be then the pool of similar questions that a particular quiz requests a problem to be selected at random for a particular student who accesses that quiz online. An example of that resulting pool of questions is illustrated in Figure 5, with titles created from the Mail Merge titles—the

section/folder name is created at the time of the iLearn upload of the Zip file. The instructor may instead simply select a different question from this pool each semester a new course is created for a given semester, or across sections.

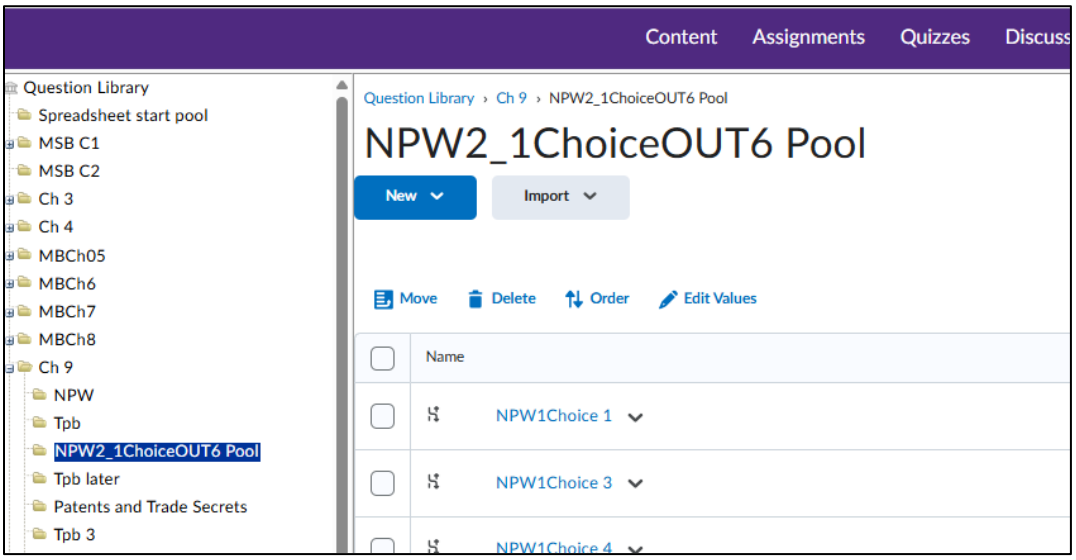


Figure 5. iLearn section/folder example of resulting upload from Respondus-generated ZIP file.

D2L allows for limited HTML content to be interpreted and displayed within the questions and feedback. The HTML may be directly edited into the original MS Word file, or even programmed for creation within MS Excel. The HTML script is identified separate from the regular text information by starting and ending labels, [HTML] … [/HTML]. A very simple example is already illustrated in the two problem statements displayed at the start of this topic. It was used to imbed an underlined sentence within the problem statement. Bold and other options are possible with HTML—they will pass through the Respondus transformation process untouched, and then are interpreted as HTML within iLearn (D2L). Figure 6 illustrates this limited example which is transformed into an iLearn preview of one of the question instances.

Graphical information, such as JPEG files, may also may be imbedded within the MS Word question statements as HTML references, though the process requires a separate uploading of the graphical content to a specific, known folder location within the LMS folder for the course. Once that folder location and the file names are established separately in the LMS, then the HTML coding may be included in both the problem statement and the feedback, as needed; the folder location and filename are included as local links for that course shell.

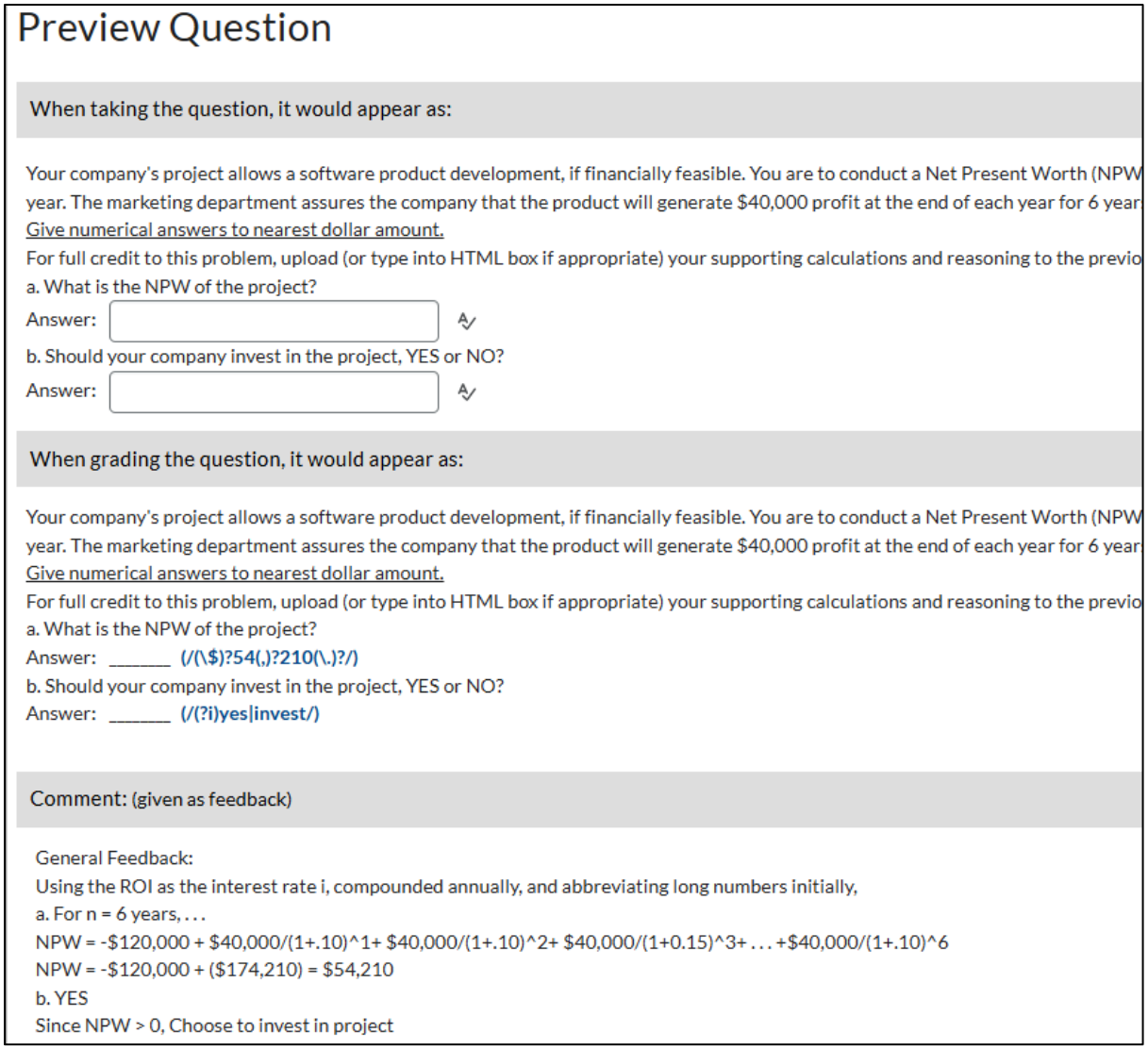


Figure 6. Example instance illustrating online result with underline formatted statement.

Tables may be manually created in the iLearn quiz tool interface. However, more elaborate tables may be created and imported quickly by first creating in Excel, coping into the iLearn quiz tool interface, then copying the HTML displayed results. Those results can then be inserted back into an iLearn question for replication using the methods noted previously. A recent drawback to this approach is that updates to iLearn have restricted the character population in the problem statement and the feedback, and elegantly formatted tables, copied from Excel, result in very large amount of text data. These types of tables may be rejected by the iLearn editing software as a result, exceeding the number of characters limit.

Though twenty variations of the question are illustrated in the example here, many more could be generated for a single question by simply increasing the number of variations in the MS Excel Qnum

column. That number of rows would translate into the same number of questions to be crated within the iLearn section.

A disadvantage of this approach is that, though regular expression formatting is created and transferred into D2L to enable automatic grading, Respondus does not establish that option to be turned on in the transfer of the ZIP file to the D2L Section; instead, the “Case Insensitive” option for each answer is set by default. Thus, the instructor must go into each individual question through the iLearn quiz tool editor and manually select “Regular Expression” for each answer. An example of this process in the D2L quiz construction interface is shown in Figure 7. Eliminating this last step may be as simple as opening up the XML ZIP file with an XML editor and accomplishing a global substitution (prior to iLearn transfer), but the author does not have a background in XML in order to investigate that option.

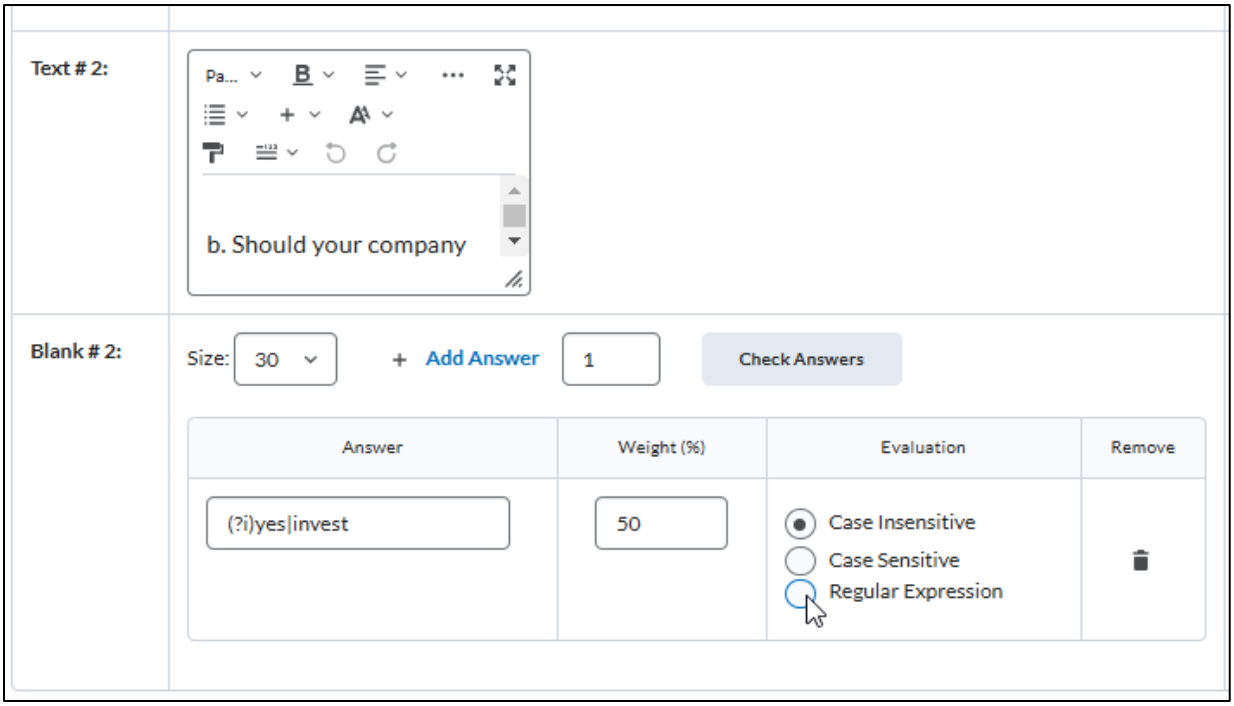


Figure 7. Flipping the switch from Case Insensitive answer to Regular Expression within iLearn Quiz Tool Editor.

As noted briefly earlier, the instructor typically provides solutions in the feedback of the question with the @ symbol after the last answer in the question is generated. Thus, a student receives the rubric within the question. This method discourages students from transferring their particular question set to a cheat-post website such as Cheggs.com, because 1) their answer may not be the only question on that topic (if random question selection is used), and 2) the feedback is inside a scrolling HTML box within their display. Thus, a pdf file is not made available for download, and the scrolling HTML box prevents detailed solutions from being printed from the webpage, as the display will often not include all the solution as visible at one time (scrolling is required).

Conclusions

This technique was adopted and modified in anticipation of large enrollment populations in the university's newly launched online and asynchronously offered engineering management masters program. However, such large enrollments have yet to materialize. There are reasons for this lack of high enrollment, but they are beyond the scope of this document. The result in the short term is that the massive number of questions in each question pool has been scaled back to no more than about 5 to 20, given the class enrollments are currently at those levels.

One issue that is repeatedly addressed by the author is the automatic grading of the regular expression phrases. The author is not steeped in programming knowledge; thus, routinely, variations in the students answers for new questions must be checked for correctness, and usually added as options in graded answers for partial credit if desired. There are many publications on large language models (LLM) (often termed a category of AI), using Python-type regular expression approaches to check such answers, even to the point of automatically providing partial credit for answers that have a diminishing distance from the correct character string for any question [16]. However, those LLM were not yet integrated into D2L by the time of this document's submission, so incorrectly answered regular expression questions must be cursorily checked by a human. Still, the answers are now embedded in the regular expressions with feedback for the specific question answers, so that a student grader may be assigned who is not necessarily an engineering management or industrial engineering student. This partial automation will likely continue until LLM language is indeed adapted into the LMS system by the university.

This non-AI, AGS method has potential for integration into an AI-powered, AGS approach, which would be supported by a future LMS; it likely will begun to be explored soon by the author as the university transitions to a new LMS that may well have AI integration. The concept would be to continue with the generation of alternative questions having specific answers, solutions, and feedback. This question data are to be fed into the AI engine, along with the student uploaded supporting information, typed into the HTML answer boxes. The goal would be for the AI assistant to not only evaluate a correct student response, but also evaluate the student's supporting information from a language model context, allowing for authorized, AI-guided, partial credit to be awarded if the student response were incorrect. Currently, the partial credit process is accomplished exclusively by the instructor.

One future imagined is that the instructor has the trained AI assistant generating questions, then the students having their own AI software answer such questions, and then the AI assistant grading these answers. How that educational process is managed by the instructor, for students and AI software, is beyond the scope of this particular publication, but will likely need addressing in the near future.

References

[1] Srikanth, M., & Asmatulu, R. (2014). Modern Cheating Techniques, Their Adverse Effects on Engineering Education and Preventions. *International Journal of Mechanical Engineering Education*, *42*(2), 129–140. https://doi.org/10.7227/IJMEE.0005

[2] Asgari, S., Trajkovic, J., Rahmani, M., Zhang, W., Lo, R. C., Sciortino, A., & Saqr, M. (2021). An observational study of engineering online education during the COVID-19 pandemic. *PloS One*, *16*(4), e0250041–e0250041. https://doi.org/10.1371/journal.pone.0250041

[3] Knight, A. M., & Nicholls, G. M., & Componation, P. J. (2012, June), Measuring the Effect of Online Homework Procedures on Student Exam Performance Paper presented at 2012 ASEE Annual Conference, San Antonio, Texas. https://peer.asee.org/21680

[4] Nicholls, Gillian M., Schell, William J.,,IV, and Neal Lewis. Best Practices for using Algorithmic Calculated Questions Via a Course Learning Management System. American Society for Engineering Education-ASEE, Atlanta, 2016. doi:https://doi.org/10.18260/p.26377.

[5] Acosta, Enrique Sanchez, and Juan Jose Escribano Otero. "Automated assessment of free text questions for MOOC using regular expressions." *Information Resources Management Journal*, vol. 27, no. 2, Apr.-June 2014, pp. 1+. Gale OneFile: Information Science, dx.doi.org/10.4018/irmj.2014040101. Accessed 25 May 2025.

[6] Xiong-Skiba, P., Buckner, S., Little, C., & Kovalskiy, A. (2020). Replacing Lab Report Grading by Online Lab Quizzes. *The Physics Teacher*, *58*(1), 55–57. https://doi.org/10.1119/1.5141975

[7] Yamani, H. A., Alharthi, A. D., & Smirani, L. K. (2022). Evaluation of Learning Management Systems: A Comparative Study Between Blackboard and Brightspace. *International Journal of Emerging Technologies in Learning*, 17(7), 125–144. https://doi.org/10.3991/ijet.v17i07.28881

[8] Goldlist, Amy, “Creating Algorithmic Quiz questions in D2L Brightspace”, (Blog), <Accessed 1/21/2026> https://amygoldlist.github.io/2021-01-15-Exams-in-D2L/

[9] Wempen, Faithe. Mail and Data Merges Using Word 2003. Que, 2008.

[10] Santoro, K., & Bilisoly, R. (2015). *Creating, Automating, and Assessing Online Homework in Introductory Statistics and Mathematics Classes*. https://doi.org/10.48550/arxiv.1501.03215

[11] Baleni, Z. G. (2012). Why and how Academic Staff Adopt e-Assessment in a Higher Education Institution (HEI). *Proceedings of the European Conference on e-Learning (Online)*, 28-36.

[12] Tan, L. Y., Hu, S., Yeo, D. J., & Cheong, K. H. (2025). A Comprehensive Review on Automated Grading Systems in STEM Using AI Techniques. *Mathematics* (Basel), 13(17), Article 2828. https://doi.org/10.3390/math13172828

[13] Pecuchova, J., Benko, Ľ., & Drlik, M. (2025). Automated Grading of Open-Ended Questions in Higher Education Using GenAI Models. International Journal of Artificial Intelligence in Education, 35(6), 3813–3846. https://doi.org/10.1007/s40593-025-00517-2

[14] Deepshikha, Deepshikha. "A Comprehensive Review of AI-Powered Grading and Tailored Feedback in Universities." Discover Artificial Intelligence, vol. 5, no. 1, 2025, pp. 251. ProQuest, https://ezproxy.tntech.edu/login?url=https://www.proquest.com/scholarly-journals/comprehensive-review-ai-powered-grading-tailored/docview/3256011427/se-2, doi:https://doi.org/10.1007/s44163-025-00517-0

[15] Mendonça, P. C., Quintal, F., & Mendonça, F. (2025). Evaluating LLMs for Automated Scoring in Formative Assessments. Applied Sciences, 15(5), Article 2787. https://doi.org/10.3390/app15052787

[16] Al Faraby, S., Romadhony, A., & Adiwijaya. (2024). Analysis of LLMs for educational question classification and generation. *Computers and Education. Artificial Intelligence*, *7*, 100298-. https://doi.org/10.1016/j.caeai.2024.100298