LEVERAGING LARGE LANGUAGE MODELS FOR THE DEVELOPMENT OF EDUCATIONAL MODULES IN MECHANICAL ENGINEERING

By

Luciano Bermudez-Viramontes

A capstone project submitted for Graduation with University Honors

May 9th, 2023

University Honors

University of California, Riverside

APPROVED

Prof. Sundararajan V

Associate Teaching Professor, Department of Mechanical Engineering Mentor

Dr. Richard Cardullo, Howard H Hays Jr. Chair

University Honors

ABSTRACT

The advent of large language models (LLMs) has revolutionized various fields, yet their application in generating educational modules for mechanical engineering still needs to be explored. This study aims to bridge this gap by developing an automated system that utilizes LLMs to create comprehensive educational content for mechanical engineering students. We employed a combination of text and image inputs to generate modules, significantly reducing the creation time from over an hour to just five minutes. The results demonstrate high accuracy and relevance in the generated content, indicating the potential of LLMs to enhance the efficiency and effectiveness of educational material development. Future work will focus on refining the system for broader applications and integrating feedback from educators to improve the modules' quality further.

ACKNOWLEDGEMENTS

I want to thank my advisor, Professor Sundar, for their continuous support, guidance, and valuable insights throughout this project. I am incredibly grateful for the opportunity to pursue this experimental and innovative project and for Professor Sundar's confidence in my work. I also want to express my gratitude to my friends and family for their help throughout the year. Their support, encouragement, and assistance have been invaluable in overcoming challenges and staying motivated.

Introduction and Background

Problem Solving in Engineering

Problem-solving, a cornerstone of an engineer's education, is a skill honed through practice. In particular, word problems are the tools educators use to teach and assess these skills, which require critical thinking and application of knowledge. Various methods are used to access word problem sets, including textbooks and online repositories. Textbooks offer a traditional approach, as they offer problem sets curated and compiled into new additions and published yearly, with some containing online learning systems that offer online modules and resources (Jackson, 2018). On the other hand, online repositories are platforms where educational materials, such as problem sets, are stored and shared and are usually compiled by educational groups, providing a wealth of problems that an instructor can access (Jackson, 2018).

Educational Resource Creation: Necessities and Constraints

Despite the benefits of these resources, they need more flexibility, which can lead to challenges in engineering education. For instance, one of the biggest challenges is that students may publish inaccurate solution sets online. Additionally, the fixed nature of the resources results in a limited variety of questions that may not align with the specific objective of the course. Given this constraint, educators play a crucial role in generating new questions. This can involve adapting from existing sources or creating entirely new content, as well as continually adapting and updating content to meet the evolving needs of their courses.

However, crafting educational content is a difficult task. Crafting problems is not merely about writing them but also ensuring that each question addresses the desired topic, is unique and error-free and that solutions are easy to follow (Jackson, 2018). Traditional methods of creating, curating, and maintaining these resources are often slow and labor-intensive, especially as the content of the course material advances. Therefore, there is a need to find a more efficient approach. Recent advances in Artificial Intelligence, particularly with generative AI and Large Language Models (LLMs), offer a promising solution to this pressing issue.

LLMs in Education: Applications and Impacts

A survey from Study.com revealed the following: 90% of students are familiar with ChatGPT (the most popular Large Language Model), with 89% stating that they have used it for homework assignments, 53% have used it to write an essay, 48% have used it for homework or quiz assistance, and 22% have used it for paper outlining. In the same survey, 82% of college professors know ChatGPT; however, only 5% have utilized it to teach a class, and 7% have used the platform to create writing prompts.

Students have already embraced the power of Large Language Models; hence, educators must grasp how to harness them for their teaching practices. The potential of LLMs goes beyond student assistance. Initiatives like Khan Academy's 'Khanmigo' have showcased the transformative power of LLMs in offering personalized tutoring (Extance, 2023). These applications leverage LLMs to probe strategic questions to students, tailoring the learning experience to their unique needs. Moreover, studies by Denny et al. (2023) have explored the quality of content generated by LLMs in learning contexts, offering valuable insights into their efficacy. The research by MacNeil et al. (2022) and Sarsa, Hellas, and Denny (2022) has honed in on applying LLMs to create course content, particularly in computer science. They discuss using LLMs to produce educational materials in computer science learning modules, including programming exercises and code explanations, further affirming its potential.

Research Gap

Despite these advancements, there has been a significant research gap; the focus has been mainly on utilizing large language models as chatbots or applications in disciplines such as computer science, while engineering education, by contrast, has seen limited integration of these tools. There has been prior research on their applications in mechanical engineering via Jackson's (2018) work, which integrated Natural Language Generation (NGL); however, this was conducted prior to the significant advancements in LLM technology and, thus, is limited by the capabilities of their time. This research, novel in its approach, intends to address this gap by examining the capabilities of LLMs to generate and tailor educational content for mechanical engineering, aiming to leverage the latest advancements to overcome previous limitations and open up new possibilities in their application.

In the ongoing work, our team has developed a software platform that integrates the science of learning with resources such as videos, interactive activities, and adaptive quizzes to improve student outcomes. However, these resources are created manually. For example, to develop a quiz question (i.e., a question where the specifics of the problem change for each student and each attempt), the content creator has to develop a set of at least four files - one each for the question/ problem stem, the code that provides the adaptive logic, the solution and the meta-data. Figures and diagrams need to be provided in additional files. In addition, if the figures have adaptive aspects (such as dimensions), the content creator has to write code separately to generate these figures adaptively. These files need to be debugged independently and upon deployment into the software platform, following which the questions need to be checked carefully to ensure that the answers and the solutions are accurately generated.

This research has developed innovative software that harnesses the power of Large Language Models and Retrieval Augmentation Generation (RAG) to create structured educational content. This content is specifically designed to streamline the organization and implementation of various engineering problems, focusing on introductory engineering courses. By improving accessibility and affordability, our software aims to close equity gaps and, most importantly, enhance student outcomes. This research will delve into the methods used to create the generator and the measures that will be used to evaluate its effectiveness.

1. **Runtime of the Software:** This refers to the total time required for the software to generate educational modules from start to finish.
2. **State of Deployment of the Modules:** This assesses the modules' readiness for practical use, including any necessary adjustments or modifications before they can be fully integrated.
3. **Correctness of the Solution:** This measures the accuracy and reliability of the content produced by the software.

Technical Background

What are Large Language Models

A large language model (LLM) is an Artificial Intelligence model that is trained on an immense amount of data, making it capable of understanding and generating natural language and other types of content to perform a wide range of tasks (IBM, 2023). The most widely known LLM is Open AI's Chat GPT-3 and GPT-4; however, there are other models, such as Meta's Llama models and PaLM 2 by Google (IBM,2023).

LLM's leverage deep learning techniques and vast amounts of textual data. These models are typically based on a transformer architecture like the generative pre-trained transformer, which excels at handling sequential data like text input (IBM,2023). They consist of multiple layers of neural networks and are trained and refined (IBM,2023). During training, the models learn to predict the next word in a sentence based on the context provided by the preceding words. The model does this by attributing a probability score to the recurrence of words that have been tokenized— broken down into smaller sequences of characters. These tokens are then transformed into embeddings, numeric representations of this context (IBM,2023). Once the model is trained, they can generate text by autonomously predicting the next work based on their input and drawing on their acquired patterns (IBM,2023). The result is a coherent and contextually relevant language generation that can be harnessed for various tasks. They can infer from context, generate coherent and contextually relevant responses, translate languages, summarize text, answer questions, and assist in writing and code generation (IBM,2023).

LLM Limitations

Despite LLM being capable of various tasks, there are still many limitations. One of the main limitations is the phenomenon of hallucination, which occurs when the generated response is incorrect while appearing valid (Extance,2023). In the context of education, this is the biggest issue plaguing LLM, as it affects their performance in solving problems accurately. There is ongoing research trying to determine how accurate these systems are. Some studies show that ChatGPT -4's performance on the SAT Math exam score was 710 out of 800, beating 89%

(Study,2023). However, there is also evidence showcasing the opposite. For example, Wei Wang, a computer scientist at UCLA, found that ChatGPT-3.5 and ChatGPT-4 got many physics-level questions wrong, where the best average score across textbooks such as Physical Chemistry, Fundamentals of Physics and Calculus Early Transcendentals, the average score was 36% (Extance,2023). Despite these issues, there are ways to limit the cases of hallucination via prompt engineering, fine-tuning, and retrieval augmentation generation.

Prompt Engineering with Large Language Models

An input is necessary to utilize a large language model called a 'prompt '. A *prompt* is a text containing instructions and information that will be fed into the LLM. The LLM will then process this prompt and generate a response. Since LLMs can understand natural language, using them is as intuitive as communicating with a person. When providing a prompt to an LLM, there are general guidelines. These include clear communication of instructions, creating a structured prompt such as defining a role, giving context data, using specific examples, and even instructing the model to evaluate its response before providing an output. Modifying and refining the prompt to achieve an ideal output is known as prompt engineering. Even minor adjustments to the prompt can significantly alter the model's response. The most basic type of prompt is zero-shot/direct prompting, which involves providing only instructions. However, there are more effective methods, such as prompting with examples of single shots (single examples) and multi-shots (multiple examples). In this approach, you present an example of an ideal output, enhancing your control over the model's response.

Fine-Tuning a Large Language Model

Prompt engineering is not the only way to specialize a model for a task; one can also fine-tune a model, where an LLM is retrained for a specific task. Fine-tuning has benefits such as improved reliability, handling edge cases, performing a task challenging to articulate in a prompt, and lower latency and cost. Fine-tuning can be very useful; however, it should be reserved for specific applications, as it requires time and effort. Before considering it, it is crucial to emphasize that achieving good results with prompt engineering is a prerequisite. Additionally, fine-tuning is inflexible as whenever a change is needed, a whole new model will need to be retrained, which is expensive and time-consuming as it requires preparation and running training, compared to prompt engineering, which provides a faster feedback loop. Finally, prompt engineering can still be used concurrently to provide even better results even if a model is fine-tuned.

Retrieval Augmentation Generation

One of the better methods in prompt engineering is providing reference texts, which allows the model to not rely on pre-trained data. This allows the model to generate more informed and up-to-date responses. This method of providing reference text is called Retrieval Augmentation Generation (RAG)**.** It allows for integrating external knowledge base, user files, and their responses instead of relying on pre-trained parameters. The main benefit of RAG is that it increases output reliability by retrieving relevant information before producing a response (Source). Additionally, there is the capability to return sources, allowing for a better understanding of where the information is gathered (Langchain, 2024). Finally, it reduces the need to fine-tune a model on new data as it can lower computational and financial costs. (Langchain, 2024)

The RAG architecture can be described as follows;

- Indexing: A method of uploading external data (Langchain, 2024)
- Retrieval: Responsible for finding relevant external data and ranking based on similarity via embeddings (Langchain, 2024)
- Generator: Use the retrieved information to generate a response (Langchain, 2024)

Methodology

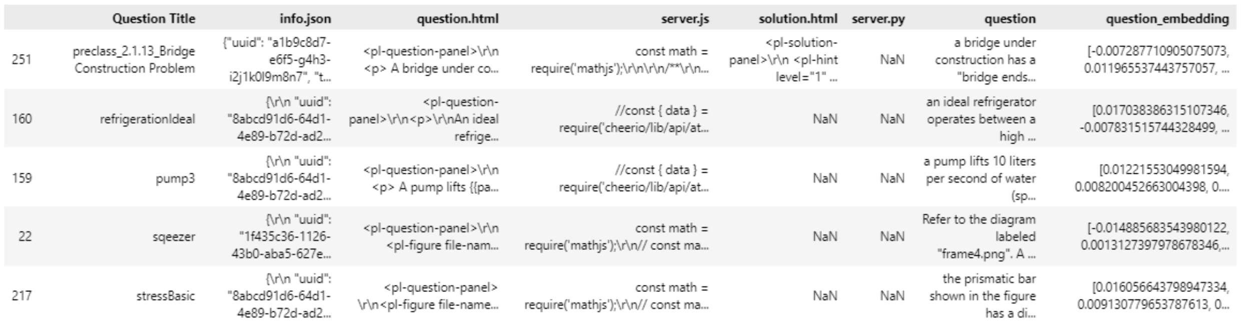
This section will outline the methodology developed to generate educational modules, showcasing the efficient transition from manual to AI-assisted techniques. Specifically, the section will detail the streamlined steps involved in preparing data and the workflow utilized for content generation. This involves harnessing the power of Large Language Models (LLMs) to facilitate content creation through prompting techniques. Furthermore, I will discuss the implementation of Retrieval-Augmented Generation (RAG) which enhances our capability by leveraging an extensive repository of examples previously curated by our team, thus significantly aiding in developing educational content. The basic structure of the generators will be discussed, and additional utilities such as image recognition and question variation generators will be discussed.

Data Preparation

This section will outline the methodology developed to generate educational modules, showcasing the efficient transition from manual to AI-assisted techniques. Specifically, the section will detail the streamlined steps involved in preparing data and the workflow utilized for content generation. These steps include data extraction, cleaning, and preparation, as well as the use of Large Language Models (LLMs) to facilitate content creation through prompting techniques. Furthermore, I will discuss the implementation of Retrieval-Augmented Generation (RAG), a technique that enhances our capability by leveraging an extensive repository of examples previously curated by our team. RAG allows us to retrieve relevant information from this repository and use it to generate new content, thus significantly aiding in developing educational content. The basic structure of the generators will be discussed, and additional utilities such as image recognition and question variation generators will be discussed.

Data Preparation

As mentioned, LLMs are very powerful, and their main benefit is their ability to generate text and code. However, we encountered some challenges during our transition into the integration of AI software. It was initially difficult to obtain an ideal output, even with more advanced prompting techniques, mainly since the structure of our modules follows a specific format. Additionally, there was the issue of each file being compatible with the others. It was noted early on that prompting by providing examples was a valid method; now, it was figuring out how to leverage this effectively. Before this research, our modules were created manually; in this process, our team, through collaborative efforts, developed a repository of 200-300 modules. These questions covered various topics, with kinematics (projectile motion, incline planes, 1-d Kinematics) being the predominant topic. Thus, we could leverage this repository to aid in content creation. To do so, the repository must be prepared, including extracting all the modules and their files and cleaning and preparing the data. Then, Open AI's embedding model is used: text-embedding-ada-002 to convert the questions into embeddings, allowing us to conduct embedding-based searches.



	Question Title	info.json	question.html	server.js	solution.html	server.py	question	question_embedding
251	preclass_2.1.13_Bridge Construction Problem	{"uuid": "a1b9c8d7-e6f5-g4h3-i2j1k0l9m8n7", "t...	<pl-question-panel>\r\n <p> A bridge under co...	const math = require('mathjs');\r\n\r\n/**\r\n...	<pl-solution-panel>\r\n <pl-hint level="1" ...	NaN	a bridge under construction has a "bridge ends...	[-0.007287710905075073, 0.011965537443757057, ...
160	refrigerationIdeal	{\r\n "uuid": "8abcd91d6-64d1-4e89-b72d-ad2...	<pl-question-panel>\r\n<p>\r\nAn ideal refrige...	//const { data } = require('cheerio/lib/api/at...	NaN	NaN	an ideal refrigerator operates between a high ...	[0.017038386315107346, -0.007831515744328499, ...
159	pump3	{\r\n "uuid": "8abcd91d6-64d1-4e89-b72d-ad2...	<pl-question-panel>\r\n <p> A pump lifts {{pa...	//const { data } = require('cheerio/lib/api/at...	NaN	NaN	a pump lifts 10 liters per second of water (sp...	[0.01221553049981594, 0.008200452663004398, 0....
22	sqeezer	{\r\n "uuid": "1f435c36-1126-43b0-aba5-627e...	<pl-question-panel>\r\n <pl-figure file-nam...	const math = require('mathjs');\r\n// const ma...	NaN	NaN	Refer to the diagram labeled "frame4.png". A ...	[-0.014885683543980122, 0.0013127397978678346,...
217	stressBasic	{\r\n "uuid": "8abcd91d6-64d1-4e89-b72d-ad2...	<pl-question-panel>\r\n<pl-figure file-name...	const math = require('mathjs');\r\n// const ma...	NaN	NaN	the prismatic bar shown in the figure has a di...	[0.016056643798947334, 0.009130779653787613, 0...

Figure 1: Overview of the Prepared Dataset of Engineering Education Modules

Using embeddings, a scatter plot (Figure 2) illustrates a cluster analysis of mechanical engineering modules used for module creation, highlighting distinct groupings in various domains. The clusters are color-coded to represent different areas, such as Mechanical Forces and Moments, Free Body Diagrams and Equilibrium, Matrix Operations and Calculus, Thermodynamics and Fluid Mechanics, and Kinematics, showcasing a visual representation of our repository

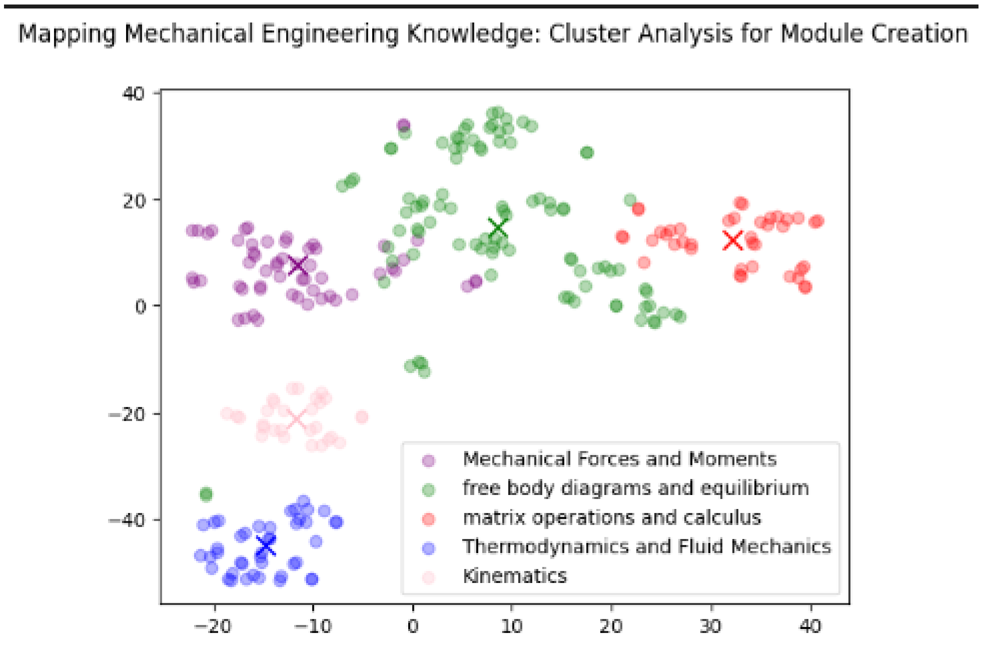


Figure 2: Cluster Analysis of Repository

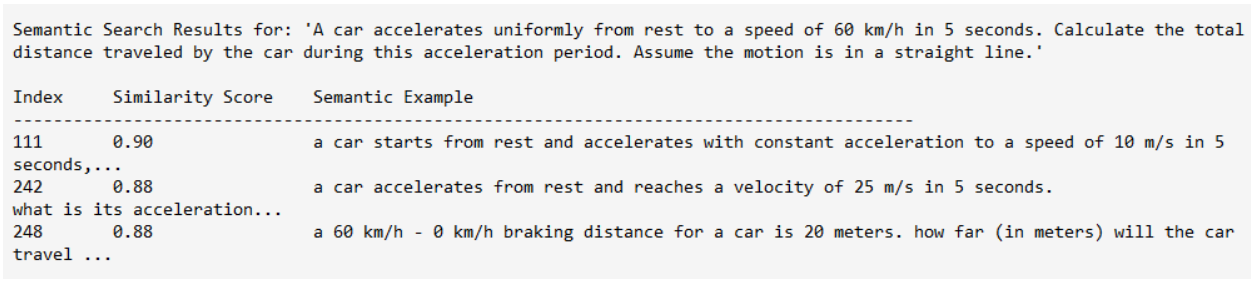
Why Leverage an Existing Repository

When we apply examples directly to the prompt, we often achieve satisfactory results. However, this approach is not without its challenges. The examples provided may not always align with our desired outcomes, leading to unexpected results. This is particularly true in the context of mechanical engineering, a broad field with diverse sub-disciplines. For instance, consider the following questions, which are typical of what mechanical engineering students might encounter.

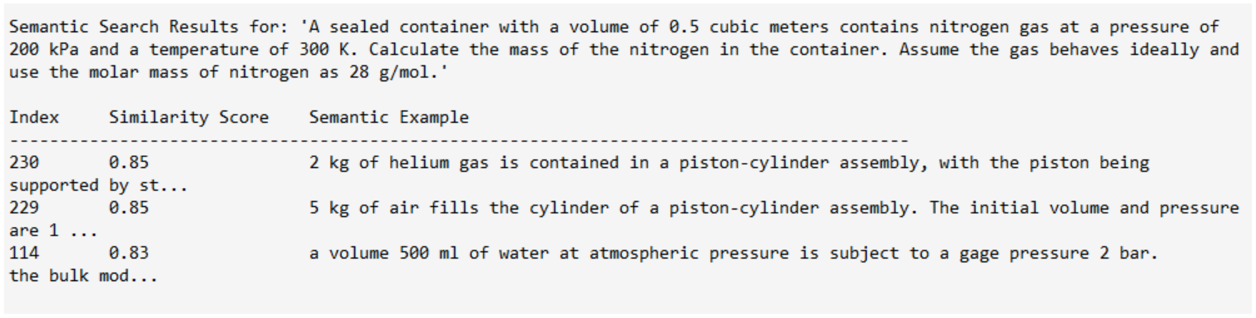
1. **Question 1: Kinematics - Acceleration and Distance:** A car accelerates uniformly from rest to a speed of 60 km/h in 5 seconds. Calculate the total distance traveled by the car during this acceleration period. Assume the motion is in a straight line.
2. **Question 2: Thermodynamics - Ideal Gas Law:** A sealed container with a volume of 0.5 cubic meters contains nitrogen gas at a pressure of 200 kPa and a temperature of 300 K. Calculate the mass of the nitrogen in the container. Assume the gas behaves ideally and use the molar mass of nitrogen as 28 g/mol.
3. **Question 3: Material Science - Stress and Strain:** A steel bar with a length of 2 meters and a cross-sectional area of 0.01 square meters is subjected to a tensile force of 3000 N. Calculate the stress and strain in the steel bar. Assume Young's modulus for steel is 200 GPa.

These represent typical questions mechanical engineering students will encounter. However, each covers a different topic and requires different considerations regarding unit conversion, answering questions, etc. Intuitively speaking, we are more likely to get an ideal response to question 1 if we provide other kinematics examples instead of thermodynamics examples; the same applies the other way around. An analysis of our repository was done shown in the foll

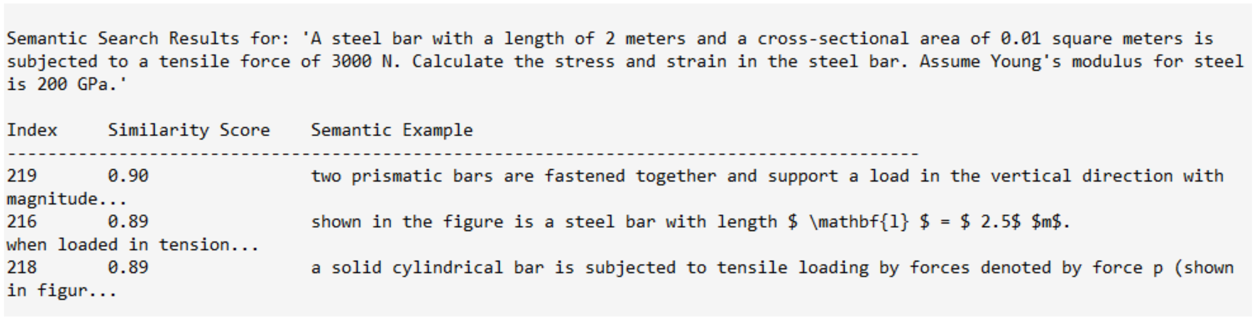
Since we have already prepared our repository, we can integrate RAG; we developed a retriever that can find similar questions in our repository by embedding the input question and returning a specified number of examples. This practical tool allows us to identify the specific question in our dataset, offers a numerical representation of the similarity, and provides the semantic example it references. This is the basic idea of the extraction. From there, we can retrieve any information that we need.



```
Semantic Search Results for: 'A car accelerates uniformly from rest to a speed of 60 km/h in 5 seconds. Calculate the total
distance traveled by the car during this acceleration period. Assume the motion is in a straight line.'

Index     Similarity Score    Semantic Example
------------------------------------------------------------------------------------------
111       0.90                a car starts from rest and accelerates with constant acceleration to a speed of 10 m/s in 5
seconds,...
242       0.88                a car accelerates from rest and reaches a velocity of 25 m/s in 5 seconds.
what is its acceleration...
248       0.88                a 60 km/h - 0 km/h braking distance for a car is 20 meters. how far (in meters) will the car
travel ...
```



```
Semantic Search Results for: 'A sealed container with a volume of 0.5 cubic meters contains nitrogen gas at a pressure of
200 kPa and a temperature of 300 K. Calculate the mass of the nitrogen in the container. Assume the gas behaves ideally and
use the molar mass of nitrogen as 28 g/mol.'

Index     Similarity Score    Semantic Example
-----------------------------------------------------------------------------------
230       0.85                2 kg of helium gas is contained in a piston-cylinder assembly, with the piston being
supported by st...
229       0.85                5 kg of air fills the cylinder of a piston-cylinder assembly. The initial volume and pressure
are 1 ...
114       0.83                a volume 500 ml of water at atmospheric pressure is subject to a gage pressure 2 bar.
the bulk mod...
```



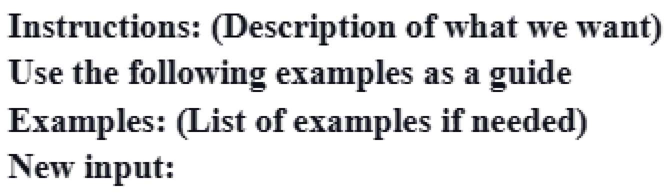
```
Semantic Search Results for: 'A steel bar with a length of 2 meters and a cross-sectional area of 0.01 square meters is
subjected to a tensile force of 3000 N. Calculate the stress and strain in the steel bar. Assume Young's modulus for steel
is 200 GPa.'

Index     Similarity Score    Semantic Example
-----------------------------------------------------------------------------------
219       0.90                two prismatic bars are fastened together and support a load in the vertical direction with
magnitude...
216       0.89                shown in the figure is a steel bar with length $ \mathbf{l} $ = $ 2.5$ $m$.
when loaded in tension...
218       0.89                a solid cylindrical bar is subjected to tensile loading by forces denoted by force p (shown
in figur...
```

Figure 3: Semantic Search of various question

Architecture of Module Generation

Each module is composed of at least four files: the question stem, the computational code, a step-by-step solution guide, and the meta-data. This necessitates the use of four distinct file generators. Three of these generators, which are based on the RAG (Retrieval-Augmented Generation), are used for the question stem, computational code, and step-by-step guide. The RAG model is particularly useful for generating these components as it combines the benefits of both retrieval and generation. The meta-data, on the other hand, does not require RAG and can be classified using the base LLM functionality.

When utilizing the software, an input question must be provided. The question is processed using OpenAI's embedding model; then, we apply an embedding search to extract similar questions from our repository. The generators follow a specific format where a structured prompt comprises specific instructions, input-output example pairs, and the new question. An example of the prompt is shown in Figure 4. Once the prompt is prepared, it is fed into either GPT-3 or GPT-4 (OpenAI, n.d.), depending on the complexity and nature of the content required. The model's temperature setting is adjusted to control the creativity and predictability of the response, allowing us to adapt to your specific needs. Putting the model temperature to a lower value allows for a more predictable outcome (OpenAI, n.d.); this is essential for our use case as it is aimed at educational content. Since each file needs to be compatible, the output of one generator is used as the input to the next to ensure they are compatible. The process generates all the needed files, including exporting them to a folder. The modules are then evaluated for accuracy by deploying and testing them on our platform. Once all the components are constructed and reviewed for quality and accuracy, they are stored and saved in our digital repository, ready to be deployed and made accessible. To visually depict the sequential steps of our content creation process, Figure 5 presents a workflow diagram illustrating each stage, from the initial input question to the final content assembly.



Instructions: (Description of what we want)
Use the following examples as a guide
Examples: (List of examples if needed)
New input:

Figure 3: General Structure of Prompt

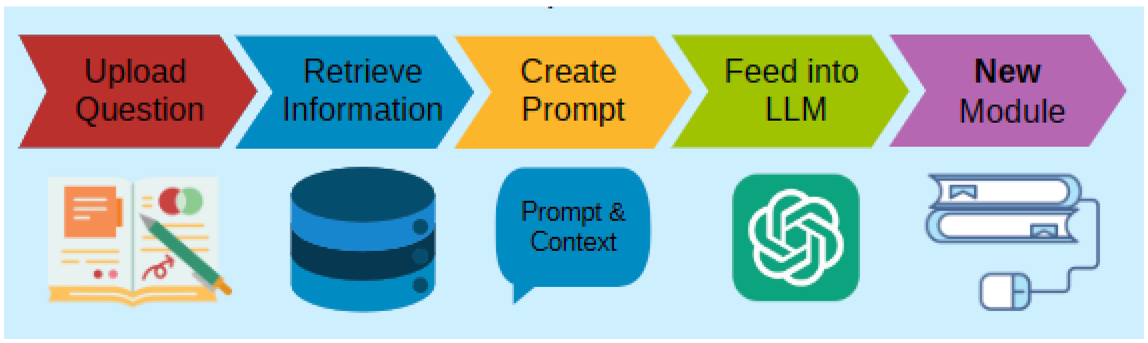


Figure 4: Workflow of Module Generation

Additional RAG

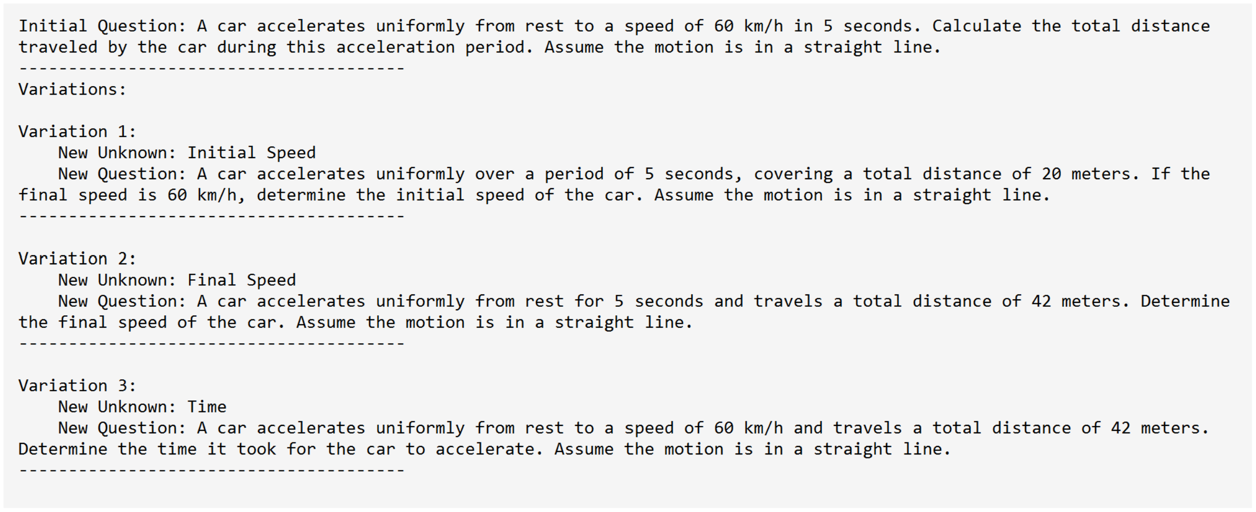
Apart from the previous examples our generators have integrated various forms of RAG, including the need for additional formatting like multiple choice options or symbolic input as answers. While our repository may not always have the correct information on implementing these considerations, we've introduced an additional retriever. This retriever uses the official documentation of the website formatting, ensuring a reliable source of information for implementation. For the code generator, which primarily generates JavaScript code, we've incorporated a retriever that reads the documentation on essential math libraries, further enhancing the code generation process.

Solution extraction via Image Input

Our current software works via a textual input or an image of the question with its corresponding solution. The textual input was the first to be implemented. However, one of the issues faced, particularly with more complex questions, was the issue of hallucination. A hallucination would cause the computational code and its solution guide to be incorrect, leading to extended debug times and essentially useless code. To combat this, we added the functionality of processing images. We add both the question and the corresponding solution guide when we provide an image. Using Image OCR, we extract both the question and the solution to the problem. The question is processed the same way as the textual; however, when we need to create computational code, we use the solution guide as a base, essentially telling it to convert the work into code. This expands the software's functionality as we are no longer limited by hallucination. The following image showcases what a usual image input into our software looks like. It is essentially handwritten notes on how to solve the given problem.

Variation Generator

One of the key functionalities of our software is the Variation Generator. This feature is designed to reformat questions and create variations from the initial input. It does this by identifying all known elements within the question and then reformulating the original question to introduce a new unknown. Importantly, this is done while ensuring that the revised question remains logical and coherent. This functionality is invaluable as it allows for the generation of diverse problem sets that maintain the educational objectives of the original query, thereby enhancing the software's adaptability.



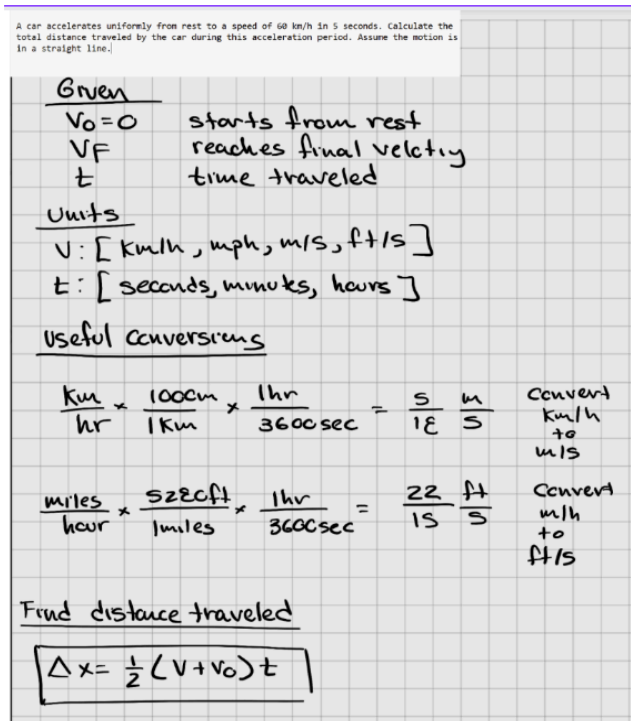
```
Initial Question: A car accelerates uniformly from rest to a speed of 60 km/h in 5 seconds. Calculate the total distance
traveled by the car during this acceleration period. Assume the motion is in a straight line.
----------------------------------------
Variations:

Variation 1:
    New Unknown: Initial Speed
    New Question: A car accelerates uniformly over a period of 5 seconds, covering a total distance of 20 meters. If the
final speed is 60 km/h, determine the initial speed of the car. Assume the motion is in a straight line.
----------------------------------------

Variation 2:
    New Unknown: Final Speed
    New Question: A car accelerates uniformly from rest for 5 seconds and travels a total distance of 42 meters. Determine
the final speed of the car. Assume the motion is in a straight line.
----------------------------------------

Variation 3:
    New Unknown: Time
    New Question: A car accelerates uniformly from rest to a speed of 60 km/h and travels a total distance of 42 meters.
Determine the time it took for the car to accelerate. Assume the motion is in a straight line.
----------------------------------------
```

Figure 3: Showcase of Variation Generator

Demonstration of Software

The following is an example of a question and its solution presented in image form, which is then inputted into the software for deployment.

Example



A car accelerates uniformly from rest to a speed of 60 km/h in 5 seconds. Calculate the total distance traveled by the car during this acceleration period. Assume the motion is in a straight line.

Given

$V_0 = 0$	starts from rest
V_F	reaches final velctiy
t	time traveled

Units

V: [km/h , mph, m/s, ft/s]

t: [seconds, minutes, hours]

Useful Conversions

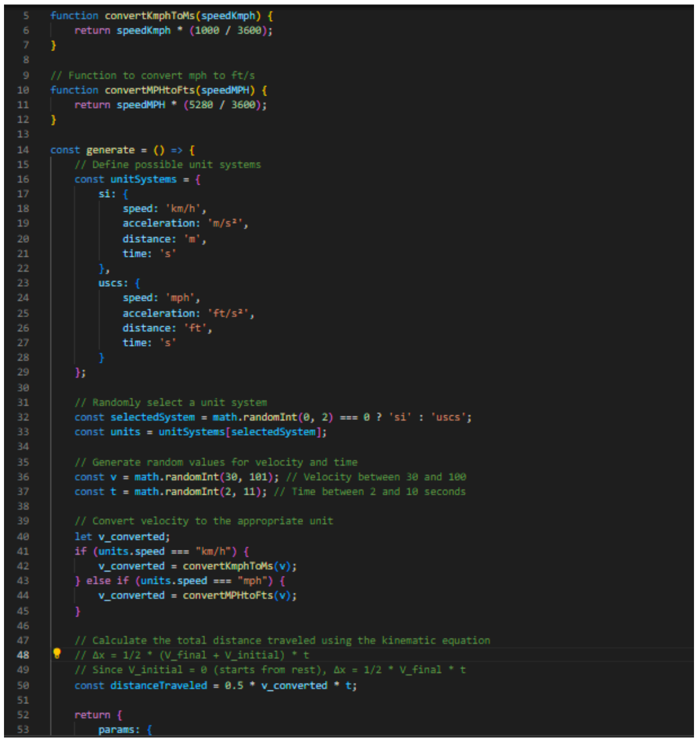
$\frac{km}{hr} \times \frac{1000m}{1km} \times \frac{1hr}{3600sec} = \frac{5}{18}\frac{m}{s}$ Convert km/h to m/s

$\frac{miles}{hour} \times \frac{5280ft}{1miles} \times \frac{1hr}{3600sec} = \frac{22}{15}\frac{ft}{s}$ Convert m/h to ft/s

Find distance traveled

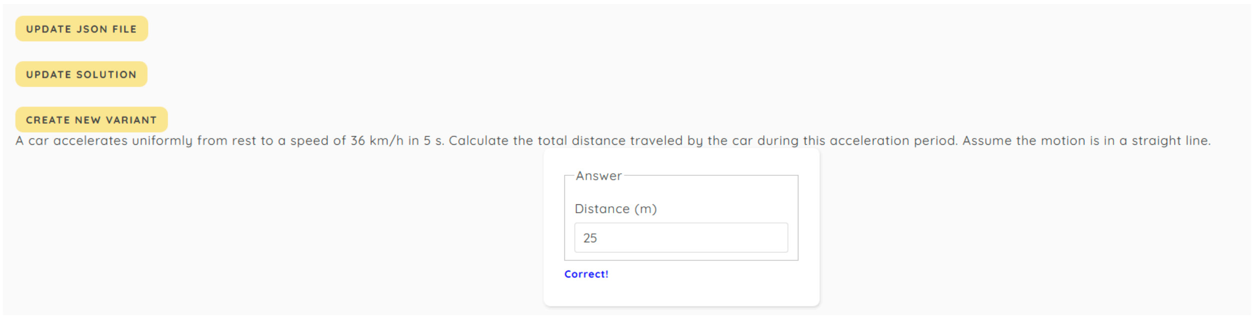
$$\boxed{\Delta x = \frac{1}{2}(V + V_0)t}$$

Figure 4: Image Input of Question



```
function convertKmphToMs(speedKmph) {
    return speedKmph * (1000 / 3600);
}

// Function to convert mph to ft/s
function convertMPHtoFts(speedMPH) {
    return speedMPH * (5280 / 3600);
}

const generate = () => {
    // Define possible unit systems
    const unitSystems = {
        si: {
            speed: 'km/h',
            acceleration: 'm/s²',
            distance: 'm',
            time: 's'
        },
        uscs: {
            speed: 'mph',
            acceleration: 'ft/s²',
            distance: 'ft',
            time: 's'
        }
    };

    // Randomly select a unit system
    const selectedSystem = math.randomInt(0, 2) === 0 ? 'si' : 'uscs';
    const units = unitSystems[selectedSystem];

    // Generate random values for velocity and time
    const v = math.randomInt(30, 101); // Velocity between 30 and 100
    const t = math.randomInt(2, 11); // Time between 2 and 10 seconds

    // Convert velocity to the appropriate unit
    let v_converted;
    if (units.speed === "km/h") {
        v_converted = convertKmphToMs(v);
    } else if (units.speed === "mph") {
        v_converted = convertMPHtoFts(v);
    }

    // Calculate the total distance traveled using the kinematic equation
    // Δx = 1/2 * (V_final + V_initial) * t
    // Since V_initial = 0 (starts from rest), Δx = 1/2 * V_final * t
    const distanceTraveled = 0.5 * v_converted * t;

    return {
        params: {
```

Figure 5: Code Snippet of Computational Code



A car accelerates uniformly from rest to a speed of 36 km/h in 5 s. Calculate the total distance traveled by the car during this acceleration period. Assume the motion is in a straight line.

Answer

Distance (m)

25

Correct!

Figure 6: Deployment of on Software

Results

This section will discuss the results obtained from applying large language models in developing educational content for mechanical engineering. While the software can automate the content creation, establishing clear benchmarks for what constitutes successful generation is of utmost importance. This rigorous process ensures that the outputs align with educational standards and effectively address the learning objectives, providing a solid foundation for content development. For this analysis, we study 20 questions from "ME002: Introduction to Mechanical Engineering." Previously administered to students, these questions are listed in the index for reference.

The first metric will be studying runtime to see how efficiently the content is generated. This comprehensive evaluation starts at the program's initiation and encompasses all user interaction, such as inputting required fields, until the end of execution. Then, the modules will be evaluated based on the deployment state, following the following criteria.

1. Minor Adjustments: These include changes to value ranges, rounding of numbers, formatting improvements, and minor code cleaning tasks. The code in this category is mainly correct and functional but needs slight tweaks to meet specific requirements.
2. Moderate Adjustments: This involves more substantial changes such as fixing unit conversions, correcting or adding new logic, or reworking parts of the code that are not optimally implemented. The overall logic and solution are correct, but sections of the code require significant review and modification.

3. Extreme Adjustments: This category includes fundamentally flawed or inefficient code requiring a complete refactor. The generated code in this state is mainly unusable and needs extensive reworking or rewriting to function correctly.

Finally, the modules will be checked based on the correctness of the solution regardless of the adjustments (minor to extreme). This is because, regardless of the adjustments made, the fundamental solution should be correct and fulfill the intended purpose. This step is crucial as it ensures that the content generated is accurate and reliable, meeting the educational standards and learning objectives.

Runtime Analysis.

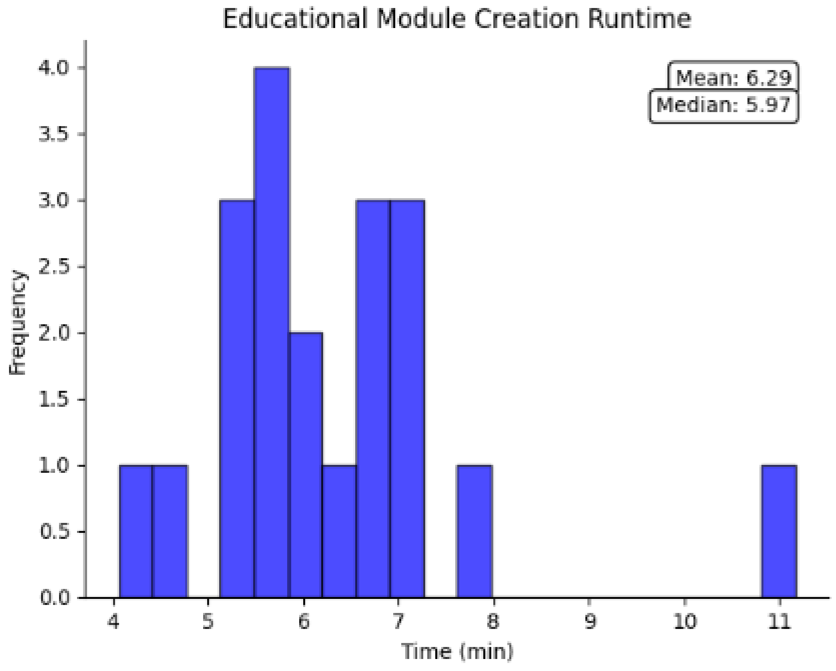
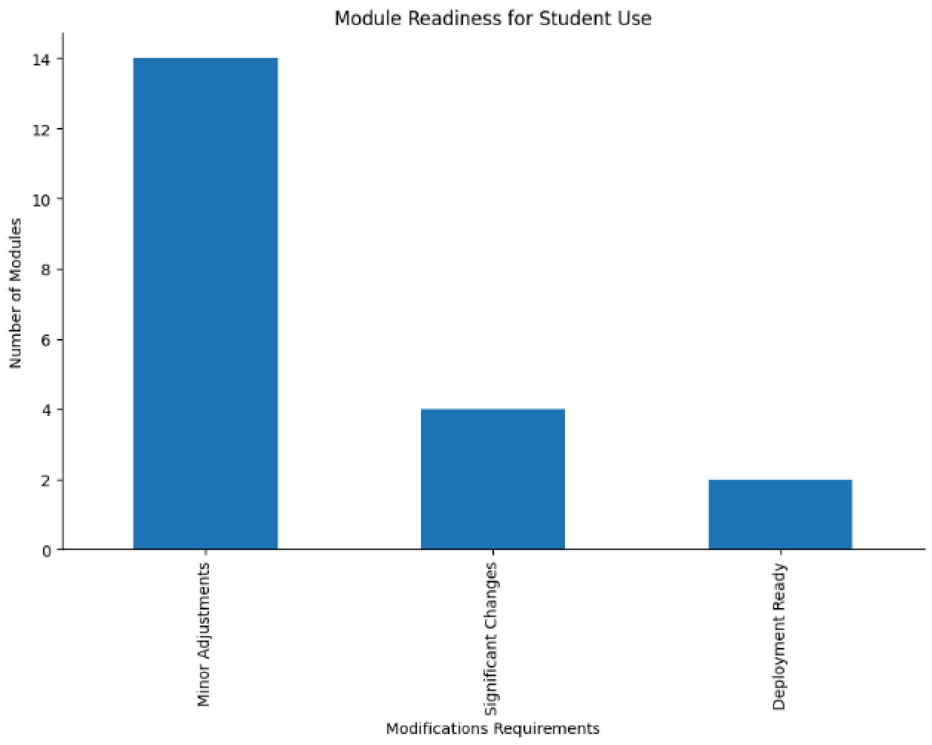
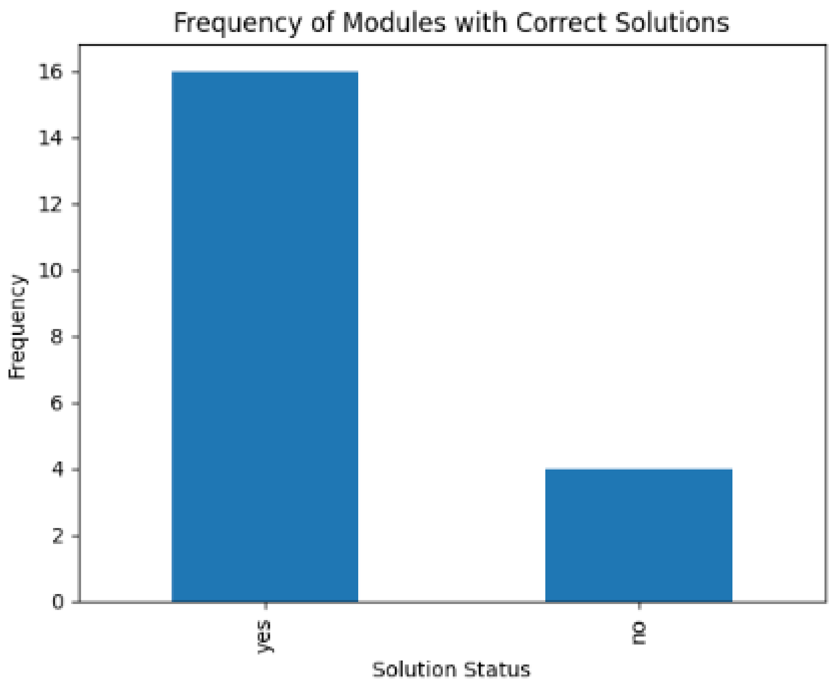


Figure 4: Runtime across module generation

Module Readiness for Deployment



Correctness of Solution



Conclusion

Efficacy and Limitations of Software

The software has demonstrated its ability in generating questions, particularly excelling in a specific type of question: engineering problems that necessitate computation, followed by a prompt for the student to respond. This approach has proven effective in our current application. However, it's important to note that this specialization also imposes limitations on the variety of question types we can generate.

Furthermore, it's important to be aware of the software's limitations. For instance, it relies on the code in our database, primarily focused on kinematics and dynamics, with less emphasis on other subjects like thermodynamics, heat transfer, and fluid dynamics. As a result, the software excels at generating more dynamics and kinematics, but may face challenges with other areas. For example the software can handle a kinematic problem without requiring external data. However, external properties are necessary for more complex engineering questions such as thermodynamics and fluid mechanics. The software incorporates the original material and a few selections in these instances. Yet, it's important to recognize that more intricate, external data-dependent questions, such as those in thermodynamics, which often involve water and steam tables, may present a challenge. This is an area where the software could potentially benefit from further development.

Improvements to the System

Our focus is on improving data retrieval and database capabilities, a key aspect in enhancing the software and developing a more intelligent system. Currently under development, the software integrates advanced retrieval augmentation techniques (RAG), like accessing code documentation to refine outputs. We are exploring additional RAG applications, including incorporating instructor materials and external functions to process external data, such as thermodynamic properties, and integrate external data more effectively. The decision to expand our training dataset from 200-300 questions to a more extensive set holds great promise, as it could significantly increase the software's diversity and performance.

Hallucinations and Question Complexity

A challenge encountered in developing this software was the occurrence of hallucinations, which caused the AI to produce inaccurate or irrelevant content. The software integrated the different text and image inputs to address this issue. Text inputs involve directly entering a question into the program. In this method, which is effective for simple questions, the AI directly determines how to solve the question and develops the files. However, complex questions lead to inaccurate responses as the model relies solely on text input without additional context or guidance. By integrating the image input, it has shown promising results. However, further research is needed to determine this threshold for question complexity.

Leveraging other LLM Features

Our system's ability to generate step-by-step solutions is a valuable resource for students. However, the implementation of LLMs as tutors also provides guidance. The potential to combine these two systems is a promising avenue. Imagine a software that accelerates module development, paired with an AI tutor that guides students to the answers. This innovative approach could significantly enhance the educational experience, offering an overall better education.

INDEX

Questions used for analysis

1. "A block of mass 2kg rests on an incline of angle. The coefficients of static and kinetic friction are 0.5 and 0.4 . It is connected via a massless string over a massless and frictionless pulley. Block has a mass of 2 kg. What's the magnitude of the acceleration of the m/s^2?",
2. A block of mass 2kg rests on an incline of angle. The coefficients of static and kinetic friction are 0.5 and 0.4 .. It is connected via a massless string over a massless and frictionless pulley. Block has a mass of 2 kg. What's the lowest mass of (in kg) that will cause the system to stay at rest?",
3. "A block of mass 2kg rests on an incline of angle. The coefficients of static and kinetic friction are 0.5 and 0.4 .. It is connected via a massless string over a massless and frictionless pulley. Block has a mass of 2 kg. What's the largest mass of (in kg) that will cause the system to stay at rest?",
4. "A wheel starts from rest and accelerates to 1800 rpm in 5 seconds. What's the angular acceleration in?",
5. "A wheel starts from rest and accelerates to 1800 rpm in 5 seconds. How many revolutions does it make in these 5 seconds? (Round to the nearest integer)",
6. "A wheel of diameter 2.5 feet is rotating at an angular speed of 1800 rpm. What's the linear speed (in ft/s) of a point on the rim of the wheel?",
7. "A wheel of diameter 2.5 feet is rotating at an angular speed of 1800 rpm. What's the centripetal acceleration (in) of a point on the rim of the wheel?"

8. "A wind turbine with an outer tip diameter of 400 feet rotates at 20 rpm. What's the linear speed (in ft/s) at the outer tip of the blades of the turbine?",
9. "A car accelerates uniformly from rest to a speed of 60 km/h in 5 seconds. Calculate the total distance traveled by the car during this acceleration period. Assume the motion is in a straight line.",
10. "A mass of 10 kg is suspended by a spring with constant 10 kN/m. By how much (in mm) does the spring extend?",
11. "A spring with spring constant 5 kN/m and initial length 5 cm is used to suspend a mass of 8 kg. What is the new length of the spring in (cm)?",
12. "A firefighter is helping a woman who weighs 120 lb. What's the tension (in lb) in the rope connecting the woman to the wall?",
13. "A rope is wrapped around a cylinder of diameter 2.5 feet so that it is wrapped around the cylinder twice completely. The rope however is a bit longer than that and the end of the rope makes an angle 60 degrees with the start of the rope. How long is the rope (in feet)?",
14. "The figure shows two carts A and B being hauled by a cable up the incline of a roller coaster. The weight of cart A is 1700 lb and the weight of cart B is 1500 lb. The carts start at the bottom of the incline and accelerate up the incline for the first 225 feet at a rate of 2ft/s^2, and then decelerate up the incline at the rate of 2ft/s^2 till they reach the top of the incline. Ignore the dimensions of the carts. What's the tension (in lbf) in the cable attached to the pulley at the top of the incline when the carts are 150 ft from the bottom of the incline?",

15. "A disk of moment of inertia 4 is initially rotating at 1200 rpm. It is brought to rest by the application of a brake. From the time the brake is engaged, the disk makes 50 rotations before it comes to a complete halt. What's the braking torque (in N-m) applied on the disk?",

16. "A car traveling at 30 mph rounds a corner with a radius of 100 ft. What is the centripetal acceleration (in ft/s^2)?",

17. "A car with weight 3000 lb traveling at 30 mph rounds a corner on an unbanked road with a radius of 100 ft. The coefficient of static friction between the car and the road is 0.8. What is the friction force (in lb) between the car and the road while cornering?",

18. "In the following figure, a force of magnitude 150 lb acts at a distance of 2 ft from the support point A. At what distance should the force of magnitude 300 lb be placed so that the moment about point A is zero?",

19. "A 5-kg block is initially at rest at the top of an incline as shown. The coefficient of kinetic friction between the block and the inclined surface is 0.2. What is the work (in J) done by gravity as it slides down to the bottom of the incline?",

20. "A disk of mass 2 kg and radius 20 cm is spinning at 300 rpm and is brought to a halt in 5 seconds. What's the torque (in N-m) required to bring the disk to rest? The moment of inertia of the disk about its axis is 1/2mr^2."

REFERENCES

Business Insider. (n.d.). GPT-4 can ace the bar, but it only has a decent chance of passing the CFA exams. Here's a list of difficult exams the ChatGPT and GPT-4 have passed. Retrieved from https://www.businessinsider.com/list-here-are-the-exams-chatgpt-has-passed-so-far-2023-1

Extance, A. (2023). ChatGPT has entered the classroom: How LLMs could transform education. *Nature*.

Google. (n.d.). Prompt Engineering for Generative AI. Retrieved from https://developers.google.com/machine-learning/resources/prompt-eng

IBM. (n.d.). What are large language models (LLMs)? Retrieved from https://www.ibm.com/topics/large-language-models

Jackson, P. (2018). Generating automated problem sets for rapid content delivery and adaptive learning modules. Paper presented at the 2018 ASEE Annual Conference & Exposition, Salt Lake City, Utah. https://doi.org/10.18260/1-2--30557

LangChain. (n.d.). Q&A with rag. 🦜🔗 Retrieved from https://python.langchain.com/v0.1/docs/use_cases/question_answering/

Sarsa, S., Denny, P., Hellas, A., & Leinonen, J. (2022). Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research* (Vol. 1). ACM.

Study.com. (n.d.). Productive Teaching Tool or Innovative Cheating? Study.com. Retrieved from https://study.com/resources/perceptions-of-chatgpt-in-schools