Investigating the Impact of Large Language Model Technologies on Syllabi and Assessment Design

Mark P. McCormack
Computer Science
Maynooth University
Maynooth, Co. Kildare, Ireland
mark.mccormack@mu.ie

John G. Keating
Computer Science
Maynooth University
Maynooth, Co. Kildare, Ireland
john.keating@mu.ie

ABSTRACT

Large Language Models (LLMs) such as ChatGPT can complete complex tasks at high quality. They can develop high-quality responses to our query if given the correct set of information and prompting. Our hypothesis is that ChatGPT may be used in a pedagogical context to develop course syllabi and assessment strategies that engage students, provide relevant experience for careers, and assist teachers in initial course development, allowing for them to focus on students' activities and queries.

For our study, we investigate the benefits that large language models provide educators regarding the generation and modification of both course syllabi and assessment strategies. We developed a set of prompts for educators to use in conjunction with existing course materials to augment and improve upon their content. We then implement these prompts with a set of Computer Science courses, tracking the differences between the original and augmented material. From this, we use GPT-generated instruments with supervision to analyze the results. These instruments assess the relevance of the material to the learning objectives and if the learning objectives are of high-quality. Our findings indicate that GPT-3 is effective in generating high-quality course syllabi and assessment strategies.

KEYWORDS

Computer Science Education, Large Language Models, ChatGPT, AI Curriculum Development, Natural Language Processing

INTRODUCTION

The key issue around LLMs in education today is that educators need tools and strategies to ensure they understand how to use LLM technologies in their pedagogy[1]. The lack of an appropriate strategies can either take educators away from more important activities while troubleshooting, or in worse cases they accept the generated material with little review which can lead to ineffective, irrelevant, or even incorrect material. As such, we require a framework for educators to use LLM tools effectively.

It is imperative then that we design strategies and prompts to make appropriate use of these technologies in the classroom, otherwise it can lead to a reduction in both course quality and student achievement. Our hypothesis is that GPT-3 can be prompted effectively to generate course syllabi and assessment strategies such that the quality of the content is high, has relevant and impactful learning objectives and the materials and assessment it creates directly correlates to these learning objectives.

Our CRTRF framework defines prompts for educators to use in five distinct stages. These prompts provide our LLM with context for what our course involves, a role that it should adopt when responding with appropriate knowledge, a detailed task that it should complete, what requirements it must obey when conducting this task and if the course content it generates should be exported in any format or style. This leaves room for exploration and modification of the framework's application for various classroom contexts.

In future iterations of our research, we hope to expand on this framework by also incorporating inter-rater reliability tests between educators and several types of LLMs. Furthermore, we would like to branch our scope outwards to generate courses for other disciplines in higher education and perhaps include more detailed prompts for specific use cases. Overall, our framework is very flexible and adaptable in its current state and can be presently used to generate high-quality course syllabi and assessments.

BACKGROUND | LITERATURE REVIEW

There is a little existing research on the applications that LLMs such as ChatGPT have in generation of course material, however it has been shown that educators across the board are skeptical or apprehensive of the technology. Murillo[2] suggests that the literature indicates educators who avoid LLM technologies in the classroom risk students not developing new skills that will help their future progress. Furthermore, this phenomenon is not new to educators as we have often had to move from our old traditional

methodologies to keep pace with current higher education demands.

In the realm of assessment generation, Speth[3] has discovered that students find LLM generated questions to be both useful in a higher education setting as well as that the questions were not obviously perceived as generated by an LLM. Furthermore, Sarsa[4] indicates that for programming-based classrooms that LLMs can generate excellent exercises that are easily influenced by teachers. This seems to indicate that LLMs have the capability of automating the assessment generation process, however our belief is that it will need proper guidance as Sarsa mentions before being considered as effective, and as such we outline a set of guidelines to do so.

There is a large amount of research that currently looks at how LLMs can be used in personalize learning experiences for students. Cao[5] shows that for an introductory programming module, when fed and trained on a large corpus of text, LLM models can assist students in their learning. This would be applicable to future iterations of our study by requiring that the material generated follow a given set of parameters for individual students. Cao's study also further proves that LLM technologies are easily influenced by educators so long as they are providing it the correct prompts.

It has been shown that LLMs can also be used for assessment grading in the classroom. This has important implications for our research as it shows that LLMs may self-assess their own learning objectives. Altamimi[6] shows that LLMs can be used to grade students affectively consistently and accurately in essays and further assessments. Therefore, this knowledge is directly applicable to our model in grading the course material. We will just need to influence its area of expertise to assess our material by the correct metrics.

METHODOLOGY

We begin by aggregating existing course metadata for five Computer Science modules at Maynooth University. These modules covered several diverse areas including Algorithms & Databases, Web Information Processing, Software Team Project, Machine Learning & Neural Networks, and Intro to Computer Science. For each module, we extracted key data from the Maynooth University course-finder including course name, learning objectives and course description. We also created columns for extra data that was not course specific, including student level (1st year Undergraduate, Masters Student, Ph.D. Student etc.), course credits (5/10/15 credits) and Continuous Assessment / Exam Mark Ratio (30% / 70%, 50% / 50% etc.). We stored this information in a CSV file. We then asked GPT-3 to analyze these pieces of data, selecting associated course name, descriptions and learning objectives and matching them with random student levels, CA/exam ratios and credit worth to generate a database of new courses we could analyze. We did this by creating a multi-step prompt that would stay focused on the activity we had planned. This framework entitled CRTRF consisted of five steps: Context, Role, Task, Requirements and Format.

Course	Module Description	Year of Study	Learning Objectives	CA / Exam Ration	Credits
CS210 (Data Structures & Algorithms)	Introduction to algorithms and data structures. Review of elementary programming concepts suitable for the implementation of abstract data types (operators, types and expressions; control of flow; methods; recursion; input & output); Algorithms for searching: linear, bounded linear and binary searches; Algorithms for sorting: selection, insertion, bubble and quick sorts; Fundamental linear data structures: stacks, queues, linked lists; Object-oriented programming: encapsulation and information hiding, classes, interfaces, class hierarchies, inheritance, polymorphism, basic exception handling; Analysis of basic algorithms.	1st Year Undergraduate	- Recognize the importance of program complexity - Describe a variety of structures for storing data such as arrays, linked lists, stacks, and queues - Explain a range of algorithms involving searching and sorting - Identify data structuring strategies appropriate to a given context - Design, develop, test, and debug object-oriented programs in Java - Apply data structuring techniques to the design of computer programs	10% / 90%	5 (125 Hours)
CS230 (Web Information Processing)	This blended-learning course provides an introduction to client-server information processing followed by an in-depth overview of the components and architecture of HTTP-based web applications. The course also provides a comparative analysis of alternative approaches to web application development using different architectures (LAMP, MEAN), frameworks, programming languages and databases. REST Web services, together with associated development technologies are explained using practical and real-world examples. All the concepts described focus on progressive enhancements (separation of structure, presentation and behaviour) and best practices in interactive, database-driven, web application development. The course also provides an overview of MVC, MVP and MVVM design practices in relation to web development.	2nd Year Undergraduate	- Describe web technologies, protocols, and architectures - Describe the difference between LAMP-like and MEAN-like architectures - Design and build a dynamic, database-driven, interactive browser-based web-based applications - Describe and use various approaches for state data management for web applications - Understand the difference between MVC, MVP, and MVVM design practices in relation to web development	20% / 80%	10 (250 Hours)

Figure 1: Course Matrix with existing details on courses and custom parameters for course credits, year of study and CA/Exam Ratio

CRTRF Framework for Educational Prompt Design
Context: This is all the information we believe LLMs should be aware of in advance of completing its activity. This includes all the existing course metadata attributes we have described above.
Role: We describe what type of person the LLM should impersonate when responding with their answer. This ensures that the response is in the appropriate tone and language, as well as narrowing the LLM's knowledge to our specific domain. e.g. "You will act as a professor of Computer Science with experience in creating your own academic modules".
Task: This is the action we want the LLM to perform. We clearly indicate the activities the LLM should engage with in a step-by-step process so it knows what order and how it should engage with the tasks.
Requirements: This optional step is used if we wanted to ensure the LLM included certain pieces of information in its response. e.g. "Please ensure that you cover the Merge-Sort algorithm in one lecture of the syllabus".
Format: This step forces the LLM to respond with the data in a particular format. This usually takes the form of certain file types; in our experiment it was JSON objects so that we could store the courses in a database. e.g. "Please provide the resulting answer in Markdown format without any accompanying text".

From the below prompt, we were able to generate 100 courses' metadata and store them for analysis. To analyze this data, we employed an analysis prompt to firstly take in the generated data and from this check if the learning objectives addressed the course syllabus effectively. To begin, we discussed with GPT-3 how it would approach assessing learning objectives for a course and it designed a five criteria framework which is as follows.

Context:

The following are the module name, module description, year of study, learning objectives, CA/Exam ratio and the number of credits for the module.

Module Name: {name}
Module Description: {description}
Year of Study: {year}
Learning Objectives: {learning_objectives}
CA/Exam Ratio: {ca_exam_ratio}
Number of Credits: {year_credits}

Role Generation:

Take a deep breath and work on this problem step-by-step. You will act as a professor of Computer Science with several years of experience teaching {name}.

Task:

As a professor teaching {name}, you are tasked with creating a 12-week course syllabi. This syllabus will include 12 detailed lecture topics, subtopics, and a detailed form of assessment for the week. You will also make sure the topic is relevant to both what areas are desired in the job market and what topics are foundational to current areas of research. Please also provide a reason for why the form of assessment you chose was selected using expert educational reasoning. You will also provide an improved course overview, five learning objectives for the course, a 12-week schedule for assignments, two full wrote sample lectures with subtopics and two sample assessment documents, where the assessment documents include the questions and tasks the students need to perform. Please be verbose in your answers.

Format:

Please provide this resulting text in a JSON format without any accompanying text. Please follow the below format for the JSON file:

"Course": "Course Name",
"Module Description": "Module Description" of type string,
"Year of Study": "Year of Study" of type string,
"Learning Objective": "Learning Objective" of type string,
"CA/ Exam Ratio": "CA/ Exam Ratio" of type string,
"Course Credits": "Course Credits" of type string,
"Course Overview": "Course Overview" of type string,
"Learning Objectives": "Learning Objectives" of type array of strings,
"Schedule": "Schedule" type array of strings,
"Lectures": "Topics and Subtopics" of type object,
"Assessments": "Assessments" of type object,
"Sample Lectures": "Sample Lectures" of type object,
"Sample Assessments": "Sample Assessments" of type object

Once again, please ensure the output is a valid JSON object with no accompanying text.

Figure 2: Custom Prompt for Generating Course Syllabi and Assessment Strategies

Upon further prompting, we were able to design in collaboration with GPT-3 a set of guidelines for how educators can use this framework in judging their course, and for us to assess how well the learning objectives of GPT-3 generated courses are. The following are the guidelines it addresses.

GPT-3.5 Generated LO Evaluation Scale
Clarity and Specificity: 0: Vague or unclear SLO with no specific focus. 1: SLO is somewhat clear but could be more specific. 2: Clear and specific SLO that clearly states what students are expected to learn.
Measurability: 0: SLO lacks a measurable action verb. 1: SLO includes a measurable verb but is still somewhat vague. 2: SLO includes a clear and measurable verb, making it easy to assess.
Achievability: 0: SLO is unrealistic or unattainable within the course context. 1: SLO is somewhat attainable but may be challenging. 2: SLO is realistic and achievable within the course.
Relevance: 0: SLO is not relevant to the course's objectives or content. 1: SLO is somewhat relevant but could be more closely aligned. 2: SLO is highly relevant and directly related to the course content.
Timeframe: 0: No specified timeframe for achieving the SLO. 1: Timeframe is vaguely defined. 2: Clear timeframe specified for achieving the SLO.

With this framework in place, we were now able to do a rudimentary evaluation of these results. Initial results indicate that these LLMs may generate rather competent learning objectives. There may be further incites in conducting this verification with multiple LLMs models to simulate inter-rater reliability, but this is an area we will study in future studies. Given we are now aware of each LO's quality, we need to assess how well the course material and lecture content lines up with the LO's. In essence, we want to verify that if the students follow along with the material in the course, that they will by virtue of keeping up achieve these LO's. For this, we used the Backward Design Framework[7], in which we ask questions of our course material to check how well it aligns with the generated learning objectives, providing supportive reasoning.

Figure 4: Design of the prompt using the Backward Design Framework alongside our custom parameters and material.

We fed GPT-3 the course syllabi and learning objectives that it designed as well as the Backward Design Framework and supervised it as it provided an evaluation of the course material and how well it matched with the learning objectives. We once again used custom prompting to ensure it had the appropriate knowledge to evaluate them and exported the results in a format

that we could easily interpret. After this evaluation we can state that there is a strong correlation between the course syllabi and learning objectives outlined by GPT-3.

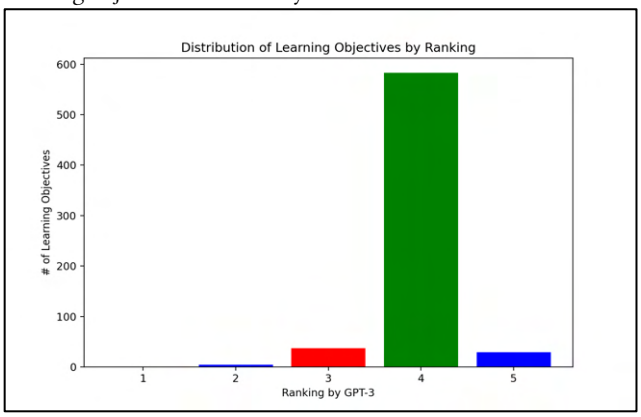

Figure 5: Evaluation of the LO Relevance to the Course Content

RESULTS

From our experiments, we can determine that GPT-3 effectively generates course syllabi and assessment strategies, as the learning objective designed by GPT-3 are achievable with the course material and assessment strategies that were generated. We've also identified that the strategies GPT-3 suggests for learning objective analysis to be rather insightful and applicable to our studies. We would like to conduct future research into this area in running inter-rater reliability tests with this framework to ensure it is effective across different LLMs and can be generalized for human use as well.

DISCUSSION

From these results we can determine that GPT-3 can be used to generate high-quality course syllabi and assessment strategies. As the literature previously suggested, this required thorough guidance to ensure GPT-3 kept on topic and had the required knowledge to complete the task correctly. Furthermore, given the fact our prompt worked on the parameters we provided, there is also room to explore integrating requirements for courses to be developed for specific audiences. This may provide teachers and students with a tool to personalize the course towards them and as such make the learning process more enjoyable.

These results also indicate that there is room to research the applicability of this framework in other fields of education. In our study we focused solely on Computer Science Education with the associated modules, however there may be variations between different subjects and as such there is plenty of room for research in the future to investigate and modify our framework to adapt to these areas.

In the future we would like to explore several extensions to this framework. Firstly, we would like to investigate inter-rater reliability by using different large language models in both our course generation and analysis phases. This will allow us to assess whether specific models have properties beneficial to education and if so, how they can be exploited. Furthermore, we would like to investigate how student profiles can be added to the requirements of our generation prompt to tailor the course towards their specific learning needs. We believe this to be beneficial as it would increase student engagement with the course given that the course will be tailored specifically to them.

Our research was limited primarily by use of a single large language model in the generation and evaluation phases. There may be more interesting results to be explored when the learning objectives are assessed by different models, several at once, or the courses are generated by a handful of different LLMs collaborating. Furthermore, we did our tests on Computer Science modules alone so there may be further results to be found when experimenting in different subject areas.

CONCLUSION

In this paper we have investigated the potential of LLMs such as GPT-3 to generate course syllabi and assessment material and found it can generate high-quality results. We have come to these conclusions by providing GPT-3 with existing course outlines and through rigorous prompt engineering requiring GPT-3 to generate several new versions of these courses for different learning levels, credits, and exam to assessment ratios. From this we assessed the quality of the generated course by looking both at the quality of the learning objectives and the course syllabi. We assess the learning objectives by using a GPT-3 generated evaluation scheme which has we have agreed upon. We then passed this scale into Llama-2 to independently analyze them and found that the learning objectives are of a high quality. We then used the Backward Design Framework with Llama-2 to analyze how well the original course materials met the requirements of the learning objective is to which they overwhelmingly did. This confirms our hypothesis that LLMs such as GPT-3 are effective in generating high-quality course syllabi and assessment strategies.

REFERENCES

[1] Montenegro-Rueda, M., Fernández-Cerero, J., Fernández-Batanero, J.M., and López-Meneses, E. 2023. Impact of the implementation of CHATGPT in education: A systematic review. *Computers 12*, 8, 153.

[2] Vargas-Murillo, A.R., Pari-Bedoya, I.N., and Guevara-Soto, F. de. 2023. The ethics of AI Assisted Learning: A systematic literature review on the impacts of CHATGPT usage in education. *Proceedings of the 2023 8th International Conference on Distance Education and Learning*.

[3] Speth, S., Meißner, N., and Becker, S. 2023. Investigating the use of AI-generated exercises for beginner and intermediate programming courses: A CHATGPT case study. *2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T)*.

[4] Sarsa, S., Denny, P., Hellas, A., and Leinonen, J. 2022. Automatic generation of programming exercises and code explanations using large language models. *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*.

[5] Cao, C. 2023. Scaffolding CS1 courses with a large language model-powered intelligent tutoring system. *28th International Conference on Intelligent User Interfaces*.

[6] Altamimi, A.B. 2023. Effectiveness of chatgpt in essay autograding. *2023 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*.

[7] Wiggins, G. and Mctighe, J. *Understanding by design, expanded 2nd Edition.* .

[8] Yang, C., Wang, X., Lu, Y., et al. 2023. Large language models as optimizers. *arXiv.org*. https://arxiv.org/abs/2309.03409.