Generative AI - Programming Assistant in the Classroom

Debasis Bhattacharyya, JD, DBA
Associate Professor
Bryson Uehara, Student
Jen Russo, Student
University of Hawaii Maui College
debasisb@hawaii.edu
maui.hawaii.edu/cybersecurity

* Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
Generative AI (GenAI), as championed by conversation chatbots like ChatGPT, has greatly impacted higher education for the past year or so. This presentation delves into the basics of Large Language Models (LLMs), prompt engineering and the impact of these technologies in the classroom.

This presentation will provide a case study about how GenAI was used in an Intermediate Programming course at the University of Hawaii Maui College in Spring 2024.

This presentation will provide the latest updates in the core features and usage of popular AI tools such as Repl.it, ChatGPT v4 from OpenAI etc.
Agenda

- Introduction
- Exploring Large Learning Models (LLMs)
- Impact of Generative AI in the Classroom
- Case Study - Using LLMs in a Programming Class
- Reflections
- Conclusions
Intelligent Machines
Broadly defined

Pattern Recognition
Learning general patterns from data

Neural Networks
Learning general patterns in unstructured data (i.e. images, text, audio, etc.)

Large Language Models
Learning to understand natural language (i.e. text)
animal didn’t cross the street because it was

Attention Is All You Need paper on Transformers, Vaswani et al. (2017)
Exploring Large Language Models (LLMs)
What if the input is an image?

**Classification**

Is it a tiger, a cat, or a fox?

---

Image classification example.

Source: Stouffelbauer, 2023
We need something way more powerful... **Neural Networks**

Neural Networks are the most powerful Machine Learning models we have today.

Source: Stouffelbauer, 2023
Language modeling

Imagine the following task: Predict the next word in a sequence

\[
\text{The cat likes to sleep in the } \_\_\_\_\_ \quad \rightarrow \quad \text{What word comes next?}
\]

Can we frame this as a ML problem? Yes, it’s a classification task.

<table>
<thead>
<tr>
<th>Word</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>ability</td>
<td>0.002</td>
</tr>
<tr>
<td>bag</td>
<td>0.071</td>
</tr>
<tr>
<td>box</td>
<td>0.085</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>zebra</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Source: Stouffelbauer, 2023
We can create **vast amounts of sequences** for training a language model.

We do the same with much **longer sequences**. For example:

A language model is a probability distribution over sequences of words. [...] Given any sequence of words, the model predicts the **next** ...

Or also with **code**:

```python
def square(number):
    """Calculates the square of a number."""
    return number ** 2
```

And as a result - the model becomes incredibly good at **predicting the next word** in any sequence.

Massive amounts of training data can be created relatively easily.

Source: Stouffelbauer, 2023
Natural language generation

After training: We can generate text by predicting one word at a time

A trained language model can

Input

LLM

LLMs are an example of what's called "Generative AI"

<table>
<thead>
<tr>
<th>Word</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>ability</td>
<td>0.002</td>
</tr>
<tr>
<td>text</td>
<td>0.084</td>
</tr>
<tr>
<td>coherent</td>
<td>0.085</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>ideas</td>
<td>0.041</td>
</tr>
</tbody>
</table>

Output at step 1

Output at step 2

Source: Stouffelbauer, 2023
What does **Generative Pre-trained Transformer (GPT)** mean?

**Generative**

Means “next word prediction.”

As just described.

**Pre-trained**

The LLM is pre-trained on massive amounts of text from the internet and other sources.

**Transformer**

The neural network architecture used (introduced in 2017).
Phases of training LLMs (GPT-3 & 4)

1. Pretraining

Massive amounts of data from the internet + books + etc.

**Question:** What is the problem with that?

**Answer:** We get a model that can babble on about anything, but it’s probably not aligned with what we want it to do.

2. Instruction Fine-tuning

Teaching the model to respond to instructions.

Model learns to respond to instructions.

→ Helps alignment

“Alignment” is a hugely important research topic

3. Reinforcement Learning from Human Feedback

Similar purpose to instruction tuning.

Helps produce output that is closer to what humans want or like.

Source: Stouffelbauer, 2023
Zero-Shot Prompting

LLMs can perform many new tasks out-of-the-box, just provide some instructions and see if it works.

Source: Stouffelbauer, 2023
Few-Shot Learning

Providing examples helps the LLM understand and follow your task.

This is especially helpful to ensure a specific output format.
Chain-of-Thought Prompting

Ask the model to solve complex tasks step by step.

Why does this work?
It gives the model a working memory, similar to humans.

User
Who won the World Cup in the year before Lionel Messi was born? Think step by step.

LLM
Lionel Messi was born on June 24, 1987. The World Cup that took place before his birth was the 1986 World Cup. The winner of the 1986 FIFA World Cup was Argentina.
Impact of LLMs to Students
1. Personalized Learning Experiences

Generative AI enables more personalized and adaptive learning experiences for students. By analyzing individual learning patterns, preferences, and performance, AI-driven platforms can tailor content, suggest resources, and adjust difficulty levels in real time. This personalization helps meet students at their level, potentially improving understanding and retention.

2. Enhanced Accessibility of Information

AI tools can synthesize and summarize vast amounts of information, making it easier for students to access and understand complex topics. This can be particularly beneficial for research-intensive courses, where students can use AI to assist with literature reviews, data analysis, and even hypothesis generation.
3. Innovative Educational Tools

Generative AI powers a range of innovative educational tools, from AI tutors that provide instant feedback to immersive simulations that offer hands-on learning experiences in virtual environments. These tools can complement traditional learning methods and provide students with additional ways to engage with the material.

4. Improved Writing and Research Skills

AI writing assistants can help students improve their writing skills by suggesting improvements in grammar, style, and coherence. Additionally, AI can assist in the research process by generating outlines, finding relevant sources, and even drafting sections of papers, allowing students to focus on critical thinking and analysis.
Case Study: Use of Coding LLMs in a Programming Course
ICS 385 - Intro to Web Programming

1. Intermediate web programming course

2. Topics
   - HTML/CSS
   - JavaScript
   - NodeJS and Express
   - EJS Templates
   - GitHub
   - RESTful APIs
   - MongoDB and Mongoose
   - Authentication and Security

3. Two student TAs
   - Took course last spring without AI
Use of Artificial Intelligence (AI)

- AI tools, such as coding generators from ReplIt, Githib Copilot, and other similar tools are welcome to be used in all classroom and course assignments, unless when specifically prohibited from doing so. Students are expected to cite and attribute their use of GenAI tools, just like any other reference tool or source, when creating their homework or exam assignments.
// This Express.js application creates two separate to-do lists, a fun to-do list and a weekend to-do list, each with pre-set arrays containing 2-3 items. The current date and a default to-do list with general items are shown on the main route ("/"). Users have the option of adding items to the work list (which is forwarded to the "/work" route) or the default list. Also, routes "/fun" and "/weekend" show listings designed for weekend chores and fun things to do.

// Preston Respicio
// 2/10/2024

const express = require("express");
const bodyParser = require("body-parser");

// create a date object that requires the date.js file
const date = require(__dirname + "/date.js");

const app = express();

// set an array for the default items in the list
let items = ["Wake up", "Brush teeth", "Get ready", "Start and warm up car", "Buy foods and drinks"];  

// set an empty array for new work items
let workItems = ["Clock-in", "Curbside check-in service", "Wheelchair assistance service"];  

// set an array for Fun and another for Weekend
let funItems = ["Watch anime episodes", "Play video games", "Walk the dog"];
```javascript
// displays index.html of root path
app.get("/", function(req, res) {
    res.sendFile(__dirname + "/index.html")
});

// invoked after hitting go in the html file
app.post("/", function(req, res) {
    // your code here
});
```

This piece of code in the `index.js` file defines how your application responds when someone visits the root URL (`/`) of your website. Let's break it down into simpler parts:

1. `app.get("/", function(req, res) {...});` This tells your web server (created with Express, a web framework for Node.js) to listen for GET requests on the root URL (`/`). A GET request is what browsers typically send when a user tries to visit a webpage.

2. `function(req, res) {...};` This is a callback function that gets executed when a user hits the root URL. It takes two arguments:
   - `req` (short for request): This object contains all the information about the request made to the server, like any data the user sends.
   - `res` (short for response): This object is used to send back a response to the user. You fill it with what you want the user to receive.

3. `res.sendFile(__dirname + "/index.html")` Inside the callback function, `res.sendFile` is used to send a file back to the user's browser. `__dirname` is a built-in variable in Node.js that gives you the absolute path of the directory containing the current executing script. So, `__dirname + "/index.html"` creates a path to the `index.html` file located in the same directory as your `index.js` script.

In summary, this piece of code serves the `index.html` page to users when they visit the root (`/`) of your website. It's like telling your web server, "Whenever someone comes to the entrance (`/`), show them the `index.html` page."
Generate code to compute the volume of a sphere

function computeSphereVolume(radius) {
    return (4/3) * Math.PI * Math.pow(radius, 3);
}

// Write test scripts to test above code to compute the volume of a sphere

function testComputeSphereVolume() {
    const radius = 5; // Example radius
    const expectedVolume = (4/3) * Math.PI * Math.pow(radius, 3);
    const calculatedVolume = computeSphereVolume(radius);
    if (calculatedVolume === expectedVolume) {
        console.log('Test Passed: Volume for radius ' + radius + ' is correctly calculated as ' + calculatedVolume + '.
    } else {
        console.error('Test Failed: Expected volume ' + expectedVolume + ', but got ' + calculatedVolume + '.
    }
}

testComputeSphereVolume();
Reflections!
**Pros**

- Design, write pseudocode
- Explain code snippets
- Generate Code, sometimes unfamiliar!
- Debug code, explain bugs
- Create Auto Test Scripts
- Helps in documentation
- Document AI section!

**Cons**

- Generate Code, sometimes unfamiliar!
- Code works but students don’t how...
- Trial and Error Approach
- Difficulty in Explaining Code
- Students who have less programming experience have more difficulty with AI
Instructor and TAs

**Pros**
- Provides another point of view for students to learn
- Students can generate working code easier
- Helps students who are unfamiliar with setup and installation over Zoom
- Students can turn to help for LLMs before asking/emails
- Students can focus on design and tests more than before!

**Cons**
- Over reliance on LLMs vs. learning by writing code
- Beginners take shortcuts
- Cheating
  - Homework Assignments
  - Exams
- Harder to give partial credit
- Unsure what students did vs. done by code generation!
- Open book exams now have images embedded in PDFs
Questions?
Comments...Discussions!

debasisb@hawaii.edu
maui.hawaii.edu/cybersecurity

* Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
Workshop Evaluation

NCyTE to add instructions / QR code

Be sure to fill out sign-in sheets before you leave!

* Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.