Generative AI is changing undergraduate education; and undergraduate research too!

Stephen MacNeil, Andrew Tran, Irene Hou
Stephen MacNeil
Assistant Professor
Temple University
Irene Hou
Incoming PhD Student
UC San Diego
Andrew Tran
Undergraduate Researcher
Temple University
How Generative AI Owns Higher Education. Now What?
Summaries of everything are easily generated, and student requirements are almost as easily satisfied. They too don't need to read or watch...

3 weeks ago

What Exactly Are the Dangers Posed by A.I.?
In late March, more than 1,000 technology leaders, researchers and other pundits working in and around artificial intelligence signed an...

May 7, 2023

Professors have a summer assignment: Prevent ChatGPT chaos in the fall
Artificial intelligence has triggered a panic among educators, who are flooding listservs, webinars and professional conferences to figure out how to deal with the technology.

August 19, 2023, 7:07 AM EDT

How Not to Be Stupid About AI, With Yann LeCun
It'll take over the world. It won't subjugate humans. For Meta's chief AI scientist, both things are true.

Dec 22, 2023
How do you feel about large language models and generative AI in computing education?

- Positive, optimistic, excited, etc.
- Negative, pessimistic, anxious, etc.

I plan to integrate / incorporate generative AI tools into my computing courses.

- Yes, absolutely
- No, definitely not

James Prather, Paul Denny, Juho Leinonen, Brett A Becker, and others. The robots are here: Navigating the generative ai revolution in computing education. (ACM ITiCSE '23)
26.1% of students who have used ChatGPT use it daily.

34.0% of students have never used it.

Hou, Irene and Mettille, Sophia and Man, Owen and Li, Zhuo and Zastudil, Cynthia and MacNeil, Stephen The Effects of Generative AI on Introductory Students’ Help-Seeking Preferences (ACM ACE ‘24)
“When using [GenAl], I don't really fully understand what they're telling me, and it's just kind of like, 'Oh! There's the answer.'

But I'm not really the one learning and really digesting what's happening and how I got to that conclusion.” (CS Student)


“As an introvert, personally, I think ChatGPT is a very good way to get help because talking to people takes away my energy”

Hou, Irene and Mettillé, Sophia and Man, Owen and Li, Zhuo and Zastudil, Cynthia and MacNeil, Stephen The Effects of Generative AI on Introductory Students’ Help-Seeking Preferences (ACM ACE ‘24)
James Prather, Paul Denny, Juho Leinonen, Brett A Becker, and others. The robots are here: Navigating the generative ai revolution in computing education. (ACM ITiCSE ’23)
"Like a Nesting Doll": Analyzing Recursion Analogies Generated by CS Students using Large Language Models. Bernstein, Seth, et al. (ACM ITICSE 2024)

"More Robots are Coming: Large Multimodal Models (ChatGPT) can Solve Visually Diverse Images of Parsons Problems." Hou, Irene, et al. (ACE 2024)

"CausalMapper: Challenging designers to think in systems with Causal Maps and Large Language Model." Huang, Ziheng, et al. (ACM C&C 2023)

"Memory sandbox: Transparent and interactive memory management for conversational agents." Huang, Ziheng, et al. (ACM UIST 2023)

"The Effects of Generative AI on Computing Students’ Help-Seeking Preferences." Hou, Irene, et al. (ACE 2024)

"Generating multiple choice questions for computing courses using large language models." Tran, Andrew, et al. (IEEE FIE 2023)

"Using large language models to automatically identify programming concepts in code snippets." Tran, Andrew, et al. (ACM ICER 2023)
**“Like a Nesting Doll”: Analyzing Recursion Analogies Generated by CS Students using Large Language Models**

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
<th>City</th>
<th>Country</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seth Bernstein</td>
<td>Temple University</td>
<td>Philadelphia</td>
<td>PA, United States</td>
<td><a href="mailto:seth.bernstein@temple.edu">seth.bernstein@temple.edu</a></td>
</tr>
<tr>
<td>Paul Denny</td>
<td>University of Auckland</td>
<td>Auckland</td>
<td>New Zealand</td>
<td><a href="mailto:paul@cs.auckland.ac.nz">paul@cs.auckland.ac.nz</a></td>
</tr>
<tr>
<td>Juho Leinonen</td>
<td>University of Auckland</td>
<td>Auckland</td>
<td>New Zealand</td>
<td><a href="mailto:juho.leinonen@ausckland.ac.nz">juho.leinonen@ausckland.ac.nz</a></td>
</tr>
<tr>
<td>Lauren Kan</td>
<td>Temple University</td>
<td>Philadelphia</td>
<td>PA, US</td>
<td><a href="mailto:lauren.kan@temple.edu">lauren.kan@temple.edu</a></td>
</tr>
<tr>
<td>Arto Hellas</td>
<td>Aalto University</td>
<td>Espoo</td>
<td>Finland</td>
<td><a href="mailto:arto.hellas@aalto.fi">arto.hellas@aalto.fi</a></td>
</tr>
<tr>
<td>Matt Littlefield</td>
<td>Temple University</td>
<td>Philadelphia</td>
<td>PA, US</td>
<td><a href="mailto:matt.littlefield@temple.edu">matt.littlefield@temple.edu</a></td>
</tr>
<tr>
<td>Sami Sarsa</td>
<td>Aalto University</td>
<td>Espoo</td>
<td>Finland</td>
<td><a href="mailto:sami.sarsa@aalto.fi">sami.sarsa@aalto.fi</a></td>
</tr>
<tr>
<td>Stephen MacNeil</td>
<td>Temple University</td>
<td>Philadelphia</td>
<td>PA, US</td>
<td><a href="mailto:stephen.macneil@temple.edu">stephen.macneil@temple.edu</a></td>
</tr>
</tbody>
</table>
“Using Harry Potter as an example for the analogy for Recursion fits well because Harry Potter has a cult following and is widely popular amongst Gen Z... it allows students to understand how recursion works given a familiar context... associating the code with a story like the Chamber of Secrets, the concept of recursion can provide a sense of engagement, which makes the recursion; a difficult concept can be grasped easier than other potential analogies.” (P277)
“Like a Nesting Doll”: Analyzing Recursion Analogies Generated by CS Students using Large Language Models

Seth Bernstein  
Temple University  
Philadelphia, PA, United States  
seth.bernstein@temple.edu

Paul Denny  
University of Auckland  
Auckland, New Zealand  
paul@cs.auckland.ac.nz

Juho Leinonen  
University of Auckland  
Auckland, New Zealand  
juho.leinonen@auckland.ac.nz

Lauren Kan  
Temple University  
Philadelphia, PA, US  
lauran.kan@temple.edu

Arto Hellas  
Aalto University  
Espoo, Finland  
arto.hellas@aalto.fi

Matt Littlefield  
Temple University  
Philadelphia, PA, US  
matt.littlefield@temple.edu

Sami Sarsa  
Aalto University  
Espoo, Finland  
sami.sarsa@aalto.fi

Stephen MacNeil  
Temple University  
Philadelphia, PA, US  
stephen.macneil@temple.edu

“Using Harry Potter as an example for the analogy for Recursion fits well because Harry Potter has a cult following and is widely popular amongst Gen Z... it allows students to understand how recursion works given a familiar context... associating the code with a story like the Chamber of Secrets, the concept of recursion can provide a sense of engagement, which makes the recursion; a difficult concept can be grasped easier than other potential analogies.” (P277)

“The reason why I love this analogy so much is it use the concept of unwrapping Nesting Doll, which was one of my favourite toys and cartoon when I was young. In fact that when I heard {Anonymized Instructor} talked about recursion, I immediately remember the Nesting Dolls. The AI successfully utilise that concept and clearly explain how the code use recursion to work which make me understand it without much problem.” (P163)
Undergraduate research gives students agency in the face of uncertainty!

- Students are not victims of Generative AI
- Students choose whether to use Generative AI
- Students adapt and reappropriate Generative AI
- Students can shape the direction of Generative AI
- Students are leading the conversation about GenAI


Tran, Andrew, et al. "Generating multiple choice questions for computing courses using large language models." (IEEE FIE 2023)
The Effects of Generative AI on Computing Students’ Help-Seeking Preferences

Irene Hou
irene.hou@temple.edu

Sophia Metille
sophia.metille@temple.edu

Zhuo Li
tuq23149@temple.edu

Owen Man
owen.man@temple.edu

Cynthia Zastudil
cynthia.zastudil@temple.edu

Stephen MacNeil
stephen.macneil@temple.edu
Help-seeking can be challenging for students.

**Socio-emotional**
- Appearing incompetent
- Rejection
- Social cost

**Decision-making**
- Knowing when to seek help
- How to evaluate feedback
- How to ask for help
Computing students seek help through resources such as:

- Instructors/TAs
- Peers
- Online

Generative AI as a new help-seeking resource
When seeking help, how does generative AI compare to peers, instructors, TAs, or the internet?
Rank the help-seeking resources you most prefer to rely on.

- Online resources (Youtube, StackOverflow, etc.)
- Course discussion forums
- TAs
- Instructors
- Friends
- ChatGPT
- GitHub Copilot
4 Common CS Tasks

- Understanding course concepts
- Writing or generating code
- Debugging
- Developing test cases

Survey (n=47) of NA university computing students

Gender of surveyed students:
- Male: 30
- Female: 10
- Did not say: 5

Student distribution across university levels:
- First year: 41.3%
- Second year: 19.6%
- Third year: 26.1%
- Fourth year: 8.7%
- Incoming first year: 4.2%
<table>
<thead>
<tr>
<th>Resource</th>
<th>Hourly</th>
<th>Daily</th>
<th>Weekly</th>
<th>Monthly</th>
<th>Never</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online search</td>
<td>9</td>
<td>24</td>
<td>11</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Friends</td>
<td>2</td>
<td>13</td>
<td>16</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>Class forum</td>
<td>2</td>
<td>10</td>
<td>14</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>ChatGPT</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>Instructor</td>
<td>1</td>
<td>6</td>
<td>14</td>
<td>17</td>
<td>9</td>
</tr>
<tr>
<td>TA</td>
<td>1</td>
<td>3</td>
<td>15</td>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td>GitHub Copilot</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>40</td>
</tr>
</tbody>
</table>
Online search and ChatGPT - most timely.

Online search and ChatGPT - less trustworthy and lower quality compared to instructors/TAs.
ChatGPT = polarizing?

26.1% of students who have used ChatGPT use it daily.

34.0% of students have never used it.
Interview (n=8) of NA university computing students

• Semi-structured interview

• Asked students to:
  ○ Draw comparisons between resources
  ○ What they valued most and least. Why?
“But if you put it [the error message] into ChatGPT, it will come up with something. Even though it may not be right, it will still be helpful, like a correct direction.” (P3)

“On Stack Overflow, if somebody asks a question, the answer that best suits the question is voted to the top of the list. With ChatGPT and Copilot, you don’t really get any of that.” (P7)
Trade-offs: convenience vs quality

“...don’t really care if [ChatGPT] is fast, I’ll wait [for a peer or TA]...the most important thing is getting it right.”

(P8, who experienced negative GPT hallucinations)

- More convenient and fast
- Lack of feedback latency
- Perceived as lower quality
- Novices were less lenient with ChatGPT compared to experts
Social Pressure

“I like going to ChatGPT more because I didn’t feel like I had to burden my peers at all.” (P7)

“As an introvert, personally, I think ChatGPT is a very good way to get help because talking to people takes away my energy.” (P5)

- Asking ChatGPT = more comfortable
- Free of social pressures
  - Reciprocity
  - Social capital
Who received the most value?

a) Those making effective help requests
b) Those viewing ChatGPT as a search engine/comprehensive resource

“[ChatGPT] is like many humans with so much information you can search through and find conclusions from.” (P5, a senior student)
Students either used **ChatGPT as a STEP in their process** or became discouraged after receiving wrong answers.

- Preferences → tied to an individual’s mental model
- Calibrated trust
“Epistemic paradox of having to know what they do not know” still remains challenging.

There is a gap in student ability to use models.
Summary

1. Students rely on ChatGPT to avoid social pressures and “burdening” peers

2. Students rely on ChatGPT because they struggle to ask for help

3. Many students struggle to rely on ChatGPT due to lack of knowledge on how to prompt models effectively

4. Students who help-seek successfully are able to leverage iteration and reformulate help requests with ChatGPT
Please help share our follow-up survey with your students!

Andrew Tran
Undergraduate Researcher
Temple University
Hi Andrew,

I was really excited to see that you applied to join our HCI lab! I was excited to hear in your application that you share our lab's vision of developing technology for social good.
Generating Diverse Code Explanations using the GPT-3 Large Language Model
Stephen MacNeil, Andrew Tran, Dan Mogil, Seth Bernstein, Erin Ross, Ziheng Huang

Code Snippet
Given Java Code:
```java
given java code:
int algorithm (int arr[], int l, int r, int x)
{
  if (r >= l) {
    int mid = l + (r - l) / 2;
    if (arr[mid] == x)
      return mid;
    if (arr[mid] > x)
      return algorithm(arr, l, mid - 1, x);
    return algorithm(arr, mid + 1, r, x);
  }
  return -1;
}
```

1. Identifying Common Mistakes

# List the common mistakes a beginner programmer might make while writing this code. How can a beginner programmer avoid each of these mistakes?

A beginner programmer might make the following mistakes while writing this code:

1. Not understanding how binary search works - Binary search is a search algorithm that searches for an element in a sorted array by repeatedly dividing the array in half and comparing the element to the middle element of the array. If the element is not found, the search continues in the lower or upper half of the array, depending on whether the element is less than or greater than the middle element, respectively.

2. Not correctly implementing the binary search algorithm - In order to correctly implement binary search, the programmer must ensure that the array is sorted before beginning the search. Furthermore, the programmer must correctly handle the case when the element is not found in the array.

3. Not using the proper data structures - Binary search can only be performed on an array. If the programmer is using a linked list, for example, binary search cannot be used.

2. Summarizing Code

# Summarize and explain the goal of this code.

The goal of this code is to find the index of a given element in an array using a binary search algorithm. The code takes in an array, the left and right indices of the array, and the element to be found as parameters. If the element is found, the code returns the index of the element. If the element is not found, the code returns -1.

3. Time Complexity

# What is the time complexity of this code and why?

The time complexity of this code is O(log n). This is because the code is using a Binary Search algorithm, which halves the search space each time it is called.

4. Create an Analogy

# Create a real world analogy based on this code snippet:

This code is similar to a treasure hunt. You start at the middle of the array and see if the value is the same as the one you are looking for. If it is not, you check to see if it is greater or less than the value and move to the next section until you find the treasure.

5. Important Concepts

# Create a bulleted list of the most important programming concepts in the following java code snippet:

- Functional decomposition
- Top-down design
- Recursion
- Time complexity
- Big O Notation
How useful are these explanations?

Students care about tasks not prompts!
Experiences from Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book

Stephen MacNeil  
Temple University  
Philadelphia, PA, USA  
stephen.macneil@temple.edu

Andrew Tran  
Temple University  
Philadelphia, PA, USA  
andrew.tran10@temple.edu

Arto Hellas  
Aalto University  
Espoo, Finland  
arto.hellas@aalto.fi

Joanne Kim  
Temple University  
Philadelphia, PA, USA  
joanne.kim@temple.edu

Sami Sarsa  
Aalto University  
Espoo, Finland  
sami.sarsa@aalto.fi

Paul Denny  
The University of Auckland  
Auckland, New Zealand  
paul@cs.auckland.ac.nz

Seth Bernstein  
Temple University  
Philadelphia, PA, USA  
seth.bernstein@temple.edu

Juho Leinonen  
Aalto University  
Espoo, Finland  
juho.2.leinonen@aalto.fi
Explanations Types

- Line-by-line
- Summary
- Concepts
E-book: Web software development Course offered by Aalto University
- Explanations rated on a 5-point Likert Scale

- **Line-by-line** explanations were *less useful for learning* than **summary** and **concept** explanations

- Explanations were less useful when students already knew what the code does
Qualitative Analysis of low-quality code explanations

1) The explanation was overly detailed and focused on mundane aspects of the code

2) The explanation was the wrong type (e.g.: a concept explanation that read more like a line-by-line explanation)

3) The explanation mixed code and explanatory text.
1) Line-by-line explanations were **most popular, but rated least useful**

2) **50%** of students who viewed the E-book viewed an explanation

3) Students who viewed an explanation viewed **3.0 (sd = 2.7)** explanations on average

4) **More students** viewed explanations as code snippets got more challenging

**Overall, students found the explanations as both relevant and useful for their learning.**
Generating Multiple Choice Questions for Computing Courses using Large Language Models

Andrew Tran, Kenneth Angelikas, Egi Rama, Chiku Okechukwu, Temple University
David H Smith IV, University of Illinois Urbana-Champaign
Stephen MacNeil, Temple University
In Java, what is the purpose of the ‘break’ statement within the ‘switch’ statement?

A) It terminates the entire program execution.
B) It skips the current iteration of a loop
C) It is used to exit a method or function.
D) It is used to exit the switch statement and continue with the code after the switch.
Prim's Algorithm is used to solve what problem?

Correct Answer (Bolded) Distractors (3):
A) Finding the lowest parent of a heap
B) Minimum spanning tree
C) Shortest path in a graph from a source
D) Sorting Integers

Prompt:
Based on the following multiple-choice question, generate three plausible distractors and one correct answer. Mark the correct answer with an (X):

LLM Generated Distractors + Answer:
A) Calculating shortest path between two nodes
B) Determining the maximum flow in a network
C) Finding the minimum spanning tree of a connected graph (X)
D) Solving the traveling salesman problem

Canterbury Question Bank
Generative Model
New Isomorphic MCQ
Datasets

Canterbury Question Bank

Finding the median value in a complete and balanced binary search tree is:

A) O(1)
B) O(log n)
C) O(n)
D) O(n^2)
E) O(n log n)

Low-level C Course

How many times "Banana" is get printed?

int main()
{
    int x;
    for(x=0; x<10; x++){
        if(x < 1){
            continue;
        } else {
            break;
        }
        print("Banana");
    }
    return 0;
}

A) 10
B) Infinite
C) 11
D) 0
Results

- GPT-4 outperformed GPT-3
  - C Course: 90% correctness (GPT-4) vs 36.7% correctness (GPT-3)
  - Canterbury: 75.3% correctness (GPT-4) vs 30.8% correctness (GPT-3)
Generative AI is Improving

Complex Questions
GPT-4 creates better MCQ questions on stems that required a more complex task

Question Stem: “What would be the performance of removeMin and insert methods in a priority queue if it is implemented by a sorted list?”

Original Answer(X) and Distractors:
A) O(1) , O(1)
B) O(1) , O(n) (X)
C) O(n) , O(1)
D) O(n) , O(n)

Negation Tasks
GPT-4 creates better MCQ questions on stems that have a negation task

Question Stem: “Which of the following is not considered a logical operator?”

Original Answer(X) and Distractors:
A) & (X)
B) &&
C) !
D) |
Generating Multiple Choice Questions for Computing Courses using Large Language Models

Andrew Tran, Kenneth Angelakis, Egi Rama, Chiku Obotuchu, Temple University
David H. Smith IV, University of Illinois Urbana-Champaign
Stephen MacNeil, Temple University
Congratulations Andrew Tran

Andrew Tran

Andrew Tran received scholarships and awards at Temple University.

Human-computer Interaction: Large Language Models

TITLITY CITED BY YEAR

Generating diverse code explanations using the gpt-3 large language model 143 2023
0 Market, A., Tran, A., Meng, G., Bernstein, A., Tran, A., Chen, B., et al.
Proceedings of the 2023 ACM Conference on Unstructured Computing Education ...

Experiences from using code-explanation generated by large language models in a web-software development in small teams 119 2023
0 Tran, A., Meng, G., Bernstein, A., Chen, B., et al.
Proceedings of the 2023 ACM Conference on Unstructured Computing Education ...

Comparing code explanations created by students and large language models 98 2023
0 Tran, A., Meng, G., Bernstein, A., Chen, B., et al.
Proceedings of the 2023 ACM Conference on Unstructured Computing Education ...

Automatically generating a learning materials with large language models 38 2022
0 Tran, A., Meng, G., Bernstein, A., Chen, B., et al.
Proceedings of the 2023 ACM Conference on Unstructured Computing Education ...

The Implications of Large Language Models for CS Teachers and Students 19 2023
0 Tran, A., Meng, G., Bernstein, A., Chen, B., et al.
Proceedings of the 2023 ACM Conference on Unstructured Computing Education ...

Prompt midstream: Mapping prompts for large language models to UI affordances 9 2023
0 Tran, A., Meng, G., Bernstein, A., Chen, B., et al.
Proceedings of the 2023 ACM Conference on Unstructured Computing Education ...

Decoding Logical Errors: A Comparative Study on Bug Detection by Students and Large Language Models 7 2024
0 Tran, A., Meng, G., Bernstein, A., Chen, B., et al.
Proceedings of the 2023 ACM Conference on Unstructured Computing Education ...

Using large language models to automatically identify programming concepts in code snippets 7 2023
0 Tran, A., Meng, G., Bernstein, A., Chen, B., et al.
Proceedings of the 2023 ACM Conference on Unstructured Computing Education ...

Generating multiple-choice questions for computer science using large language models 3 2023
0 Tran, A., Meng, G., Bernstein, A., Chen, B., et al.
Proceedings of the 2023 ACM Conference on Unstructured Computing Education ...

Coauthors

Elaine Hwang
Assistant Professor, Temple University

Evan Freeman
School of Computing Science, Simon Fraser University

Sam Bower
Assistant Professor, New York University

Nimrod A. Zmora
Assistant Professor, New York University

Ali Haddad
Assistant Professor, New York University

Julia Lavid
Assistant Professor, New York University

Paul Denny
Assistant Professor, University of Texas at Austin

Benjamin A. Gilman International Scholarship
Generative AI appears to be **widening not narrowing** the educational divide!
Generative AI appears to be widening not narrowing the educational divide!

What might be causing this divide?

- Lack of AI literacy
- Lack of metacognitive skills
- Lack of foundational computing skills
So at least some of the **doom and gloom** is warranted?
What are we left with when Generative AI can write better code than our students?
What are we left with when Generative AI can write better code than our students?

Complexity and Uncertainty
What are we left with when Generative AI can write better code than our students?

Complexity and Uncertainty

Undergraduate Researchers are Ready
Student Engagement

Photos of Studio Time

Open House

Fun Activities
The HCI Lab at Temple University
Student Engagement

Photos of Studio Time

Open House
Papers presented today


Tran, Andrew, et al. "Generating multiple choice questions for computing courses using large language models." (IEEE FIE 2023)
Additional papers about Gen AI from undergraduates in our lab


Hou, Irene, et al. "More Robots are Coming: Large Multimodal Models (ChatGPT) can Solve Visually Diverse Images of Parsons Problems." (ACE 2024)


Student Engagement

Photos of Studio Time

Open House