Strategies and Exercises for Assessing Programming Skills at Scale

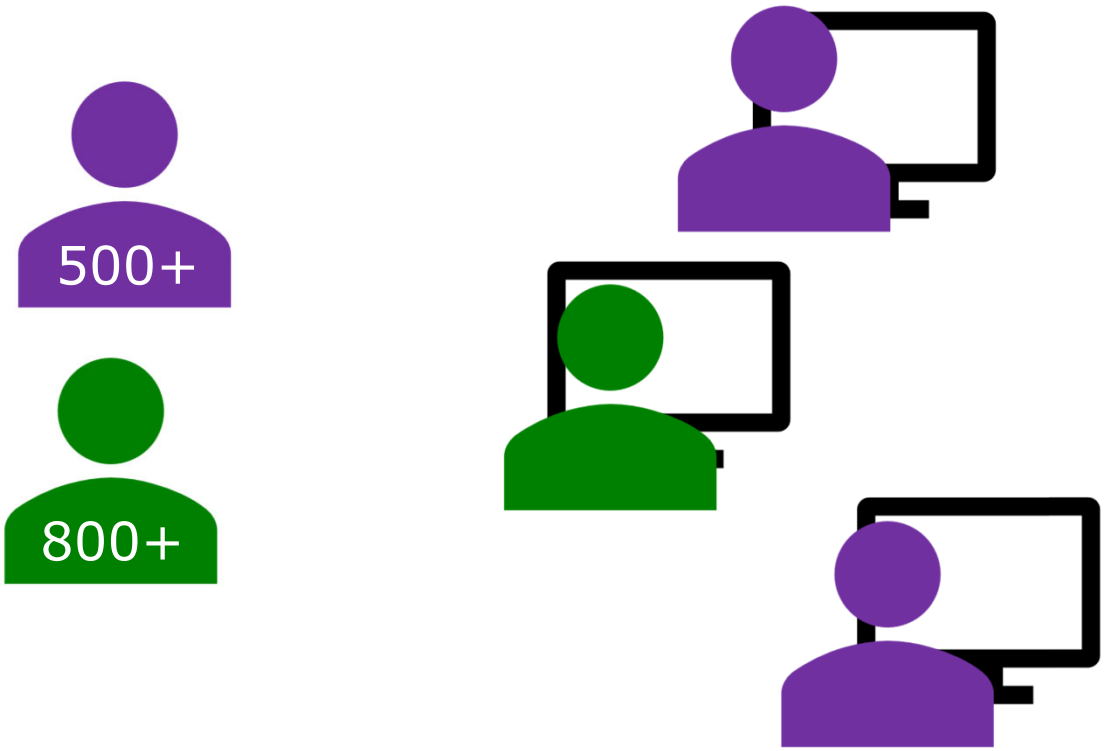
Max Fowler

Computer Science, University of Illinois

GRAINGER ENGINEERING

COMPUTER SCIENCE

Large scale, computer-based testing



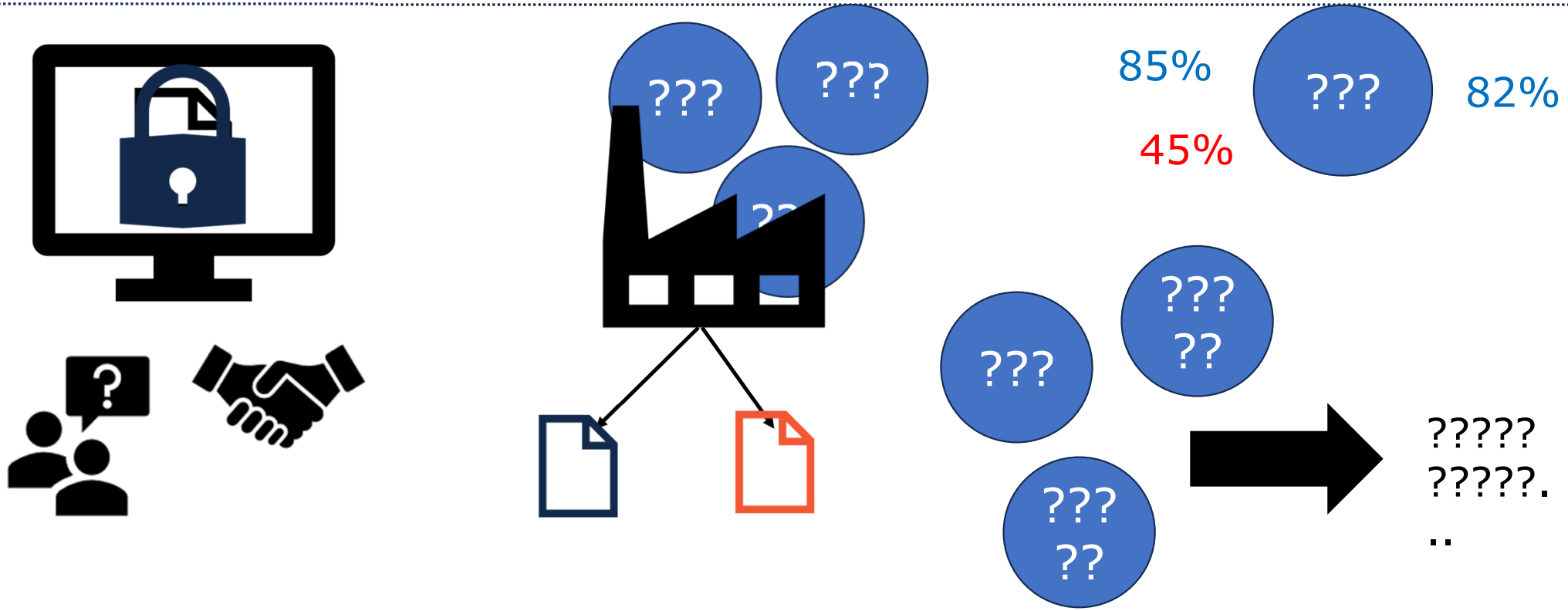
Versioned

Autograded

Async

More "authentic"

Assessments



(some) Research interests

Investigating exams and frequency

Skill hierarchies & programming patterns

Producing "isomorphic" programming questions

Autograding Explain in Plain English (EiPE)

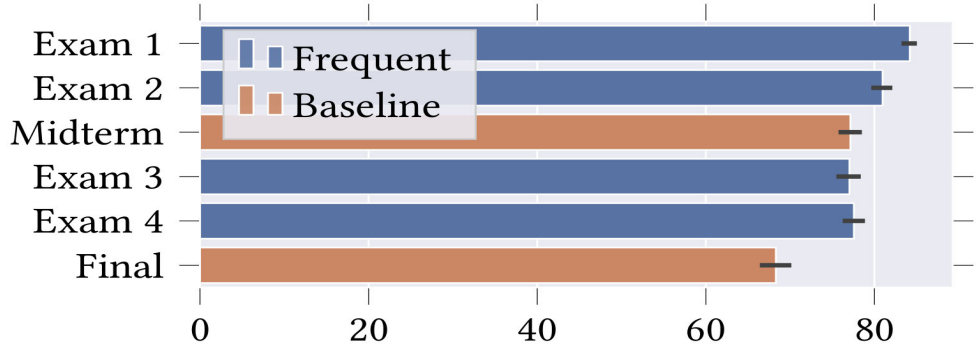
Interest one– Exams and Frequency

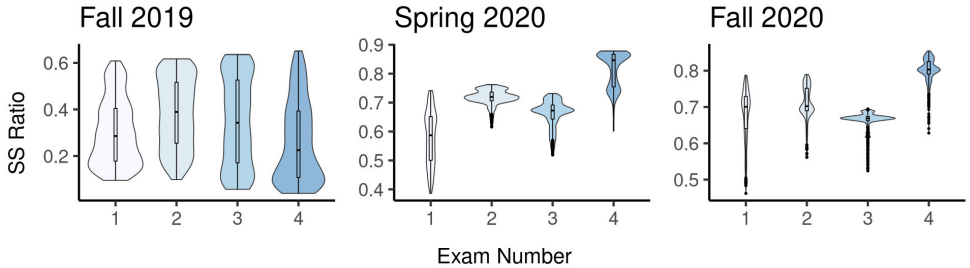
Investigating exams and frequency

Skill hierarchies & programming patterns

Producing "isomorphic" programming questions

Autograding Explain in Plain English (EiPE)





SIGCSE 2022 - "Are we Fair? Quantifying Score Impacts of Computer Science Exams with Randomized Question Pools."

SIGCSE 2023 - "Investigating the Effects of Testing Frequency on Programming Performance and Students' Behavior."

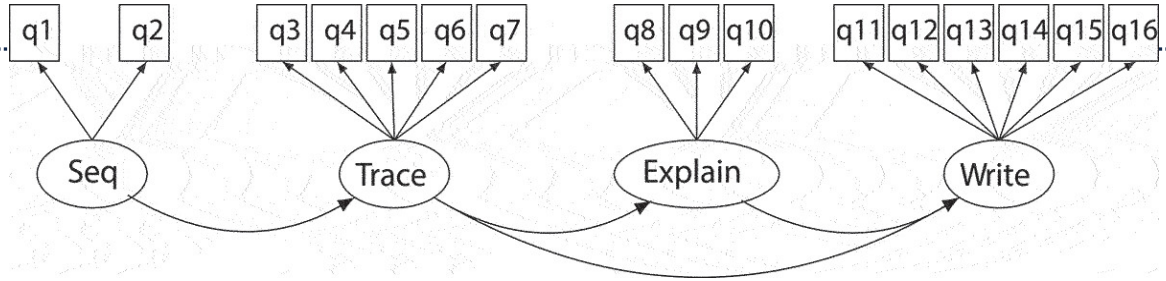
Interest two– Skills & patterns

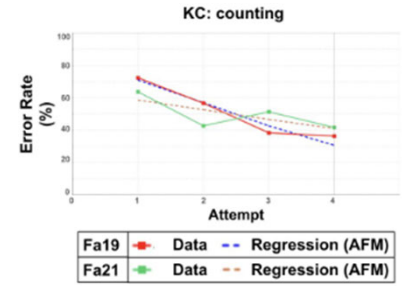
Investigating exams and frequency

Skill hierarchies & programming patterns

Producing "isomorphic" programming questions

Autograding Explain in Plain English (EiPE)






```
def count_over_100(number_list):
    count = 0
    for num in number_list:
        if num > 100:
            count += 1
    return count
```

```
def count_string_vals(a_dict):
    count = 0
    for val in a_dict.values():
        if type(val) == str:
            count +=1
    return count
```

Reevaluating the relationship between explaining, tracing, and writing skills in CS1 in a replication study. Computer Science Education Replication Studies Special Edition 2022.

Demirtas, Mehmet Arif, et al. "Validating, Refining, and Identifying Programming Plans Using Learning Curve Analysis on Code Writing Data." Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1. 2024.

Interest three– "isomorphic" questions

Investigating exams and frequency

Skill hierarchies & programming patterns

Producing "isomorphic" programming questions

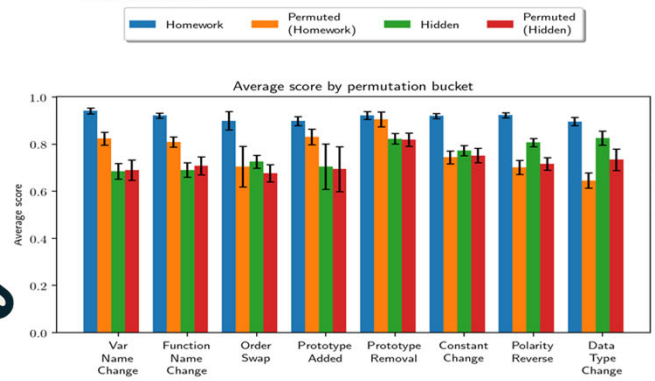
Autograding Explain in Plain English (EiPE)

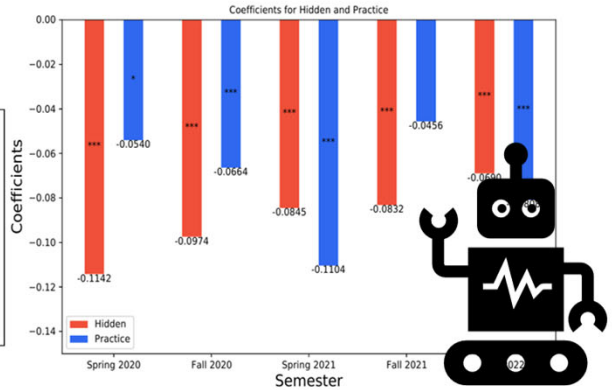
Return the sum of the last three elements of a list

Create a function called `sum_last_three` that takes a single argument of type list of numbers. Your function should return the sum of the last three elements of the given list. You can assume that the list always has at least three elements.

Return the sum of the first five elements of a list

Create a function called `sum_first_five` that takes a single argument of type list of numbers. Your function should return the sum of the first five elements of the given list. You can assume that the list always has at least five elements.





SIGCSE 2021 - "Superficial Code-guise: Investigating the Impact of Surface Feature Changes on Students' Programming Question Scores."

ITiCSE 2024 - "Quickly Producing 'Isomorphic' Exercises: Quantifying the Impact of Programming Question Permutations and Prior Exposure on Students' Performance."

Interest four– Autograding EiPE Questions

Investigating exams and frequency

Skill hierarchies & programming patterns

Producing "isomorphic" programming questions

Autograding Explain in Plain English (EiPE)

Code Reading Problem

Write a short, high-level English language description of the code in the highlighted region. *Do not give a line-by-line description.*

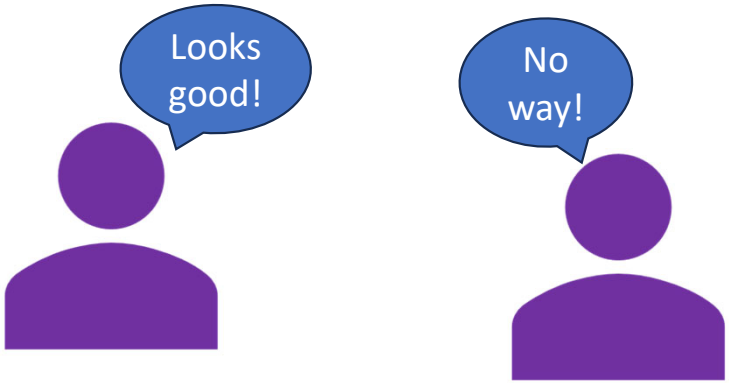
Assume that the variable x is a list of integers and variable y is an integer. *You can assume that the code compiles and runs without error.*

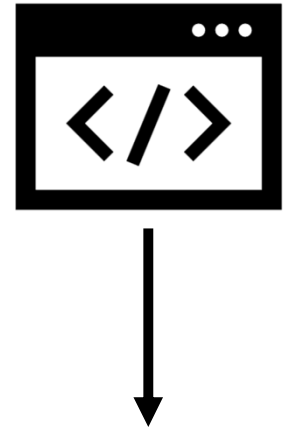
```
def f(x, y):
    for val in x:
        if val == y:
            return True
    return False
```

Returns whether a given list contains a given value

EiPE is difficult to grade

Returns whether a given list contains a given value



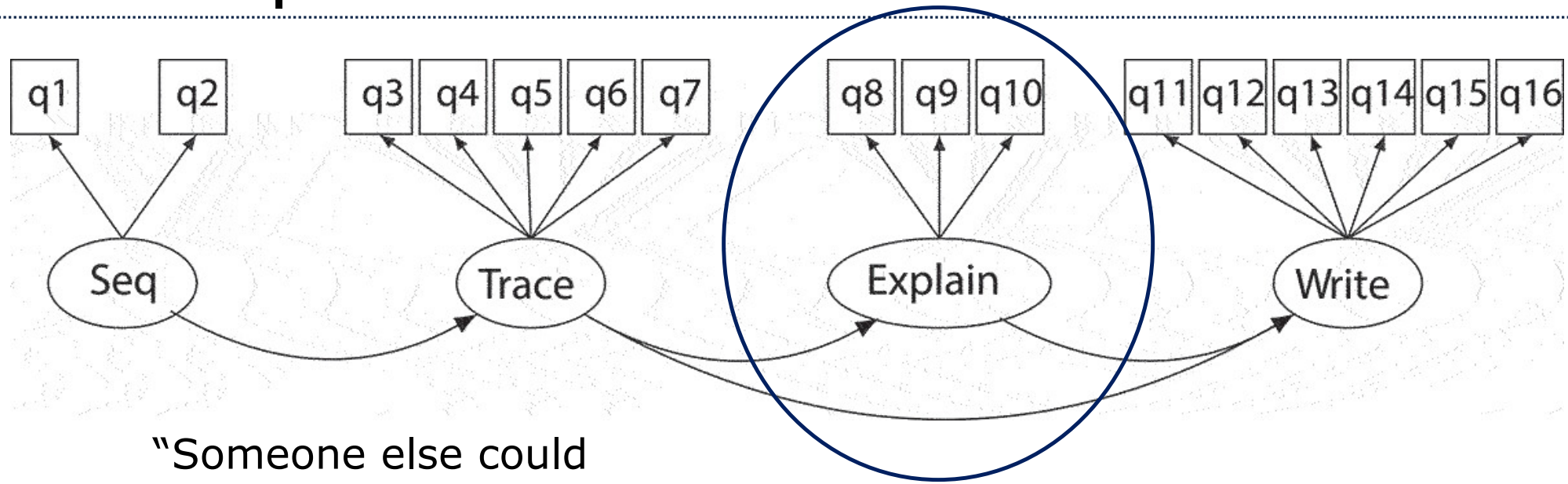


Test cases

Returns whether a given list contains a given value

???

EiPE questions assess comprehension and abstraction



"Someone else could go write the code"

Perhaps even more important with Large Language Models

Autograding pipeline (2021)

Assume that the variable x is a list of integers and variable y is an integer. *You can assume that the code compiles and runs without error.*

```
def f(x, y):
    for val in x:
        if val == y:
            return True
    return False
```

Return whether a given list contains a given value

Submitted answer

Submitted at 2023-11-07 11:34:45 (CST) 100%

hide

Return whether a given list contains a given value

Basic pre-processing

Bag-of-words & Bigram

Logistic Regression

Fowler et al. Autograding "Explain in Plain English" questions using NLP. SIGCSE 2021

Training and evaluating the grader

Returns whether a given list contains a given value

Cohen's κ = 0.83

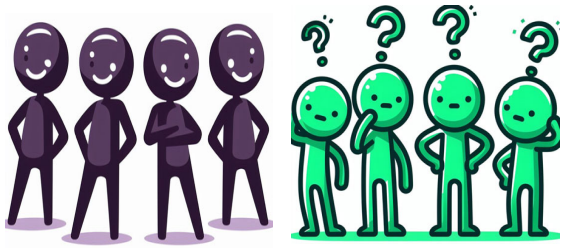
Fall 2019

Spring 2020

Ground truth

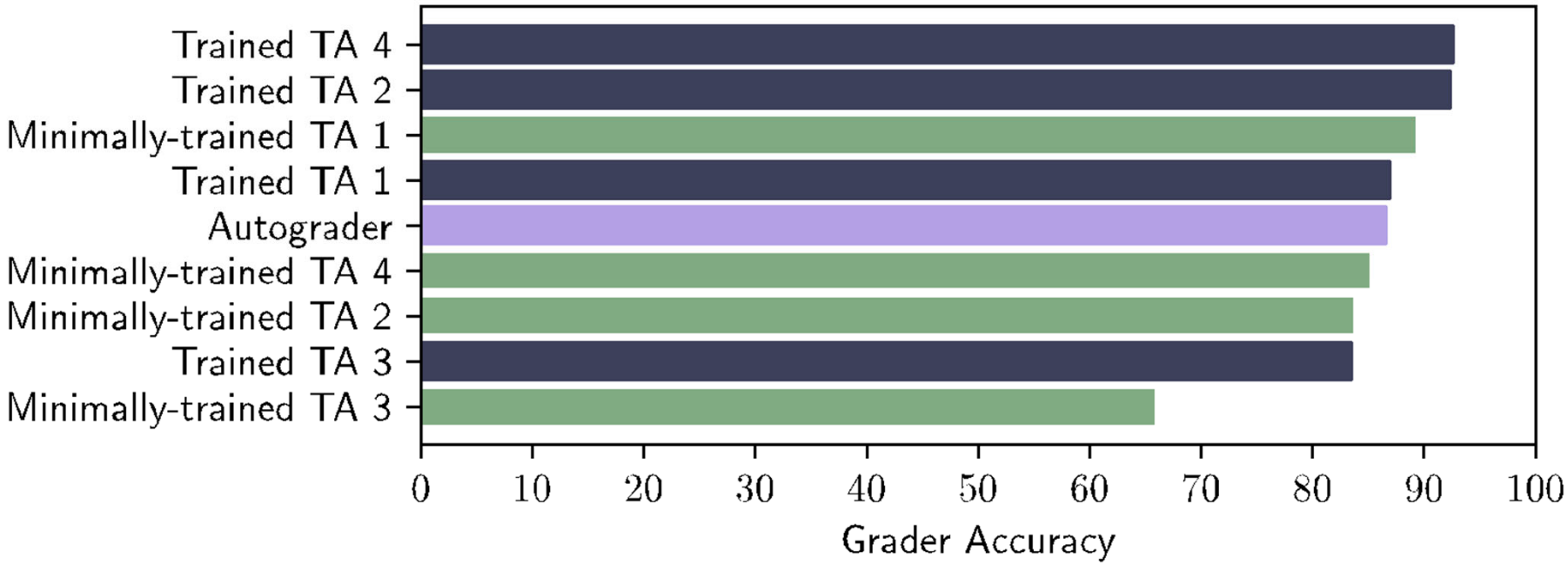
VS

Teaching assistants



Fowler et al. Autograding "Explain in Plain English" questions using NLP. SIGCSE 2021

Autograder performance vs Human TAs

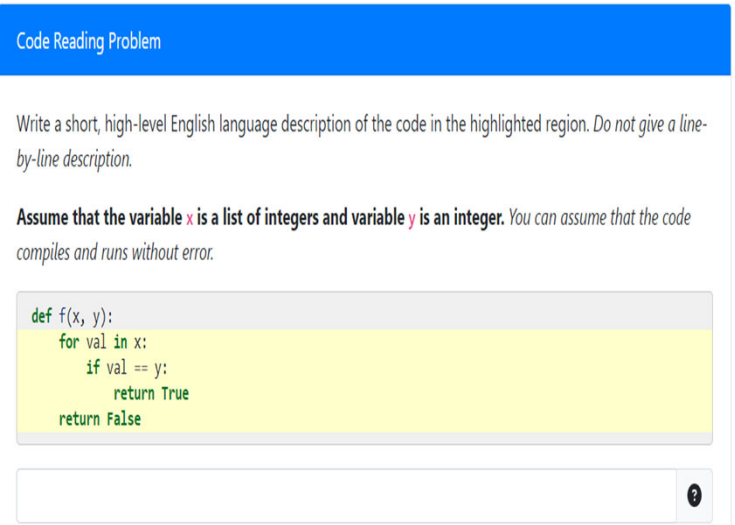


The autograder performs as well as the average TA

Fowler et al. Autograding "Explain in Plain English" questions using NLP. SIGCSE 2021

Comparing multiple autograders

Binary (Logistic, SVM)



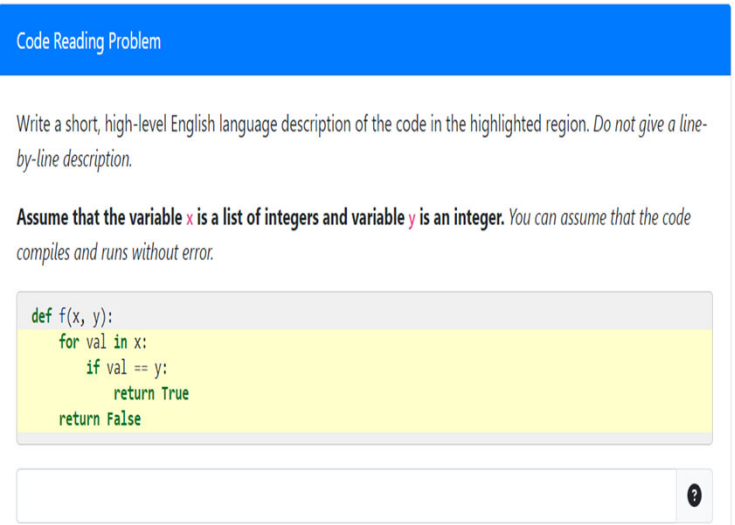
- Used the same logistic regression pipeline as before

- Also compared vs:
- SBERT SVMs
- OpenAI Embedding SVMs

Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

Comparing multiple autograders

Binary (Logistic, SVM)



Three "stacked" models – 3D

```
def f(x):
    z = []
    for y in range(x+1):
        z.append(y)
    return z
```

A desired answer: Return a list containing numbers from 0 to x

Ambiguous answer: Add the number between 0 and x to a list

Explanation: not clear that it is a new or existing list. Also, the singular "number" is confusing

```
def f(x):
    y = []
    for val in x:
        if val > 0:
            break
        y.append(val)
    return y
```

A desired answer: Makes a copy of a list up to the first positive number

Incorrect answer: Returns a list of only negative numbers from x

Explanation: indicates the wrong subset of the list

```
def f(x, y):
    if x < y:
        print(x, y)
    else:
        print(y, x)
```

A desired answer: Print two given numbers in numeric order

Low-level answer: Prints the y value then the x value if the x value is greater than or equal to y or prints the x value then the y value if x is smaller

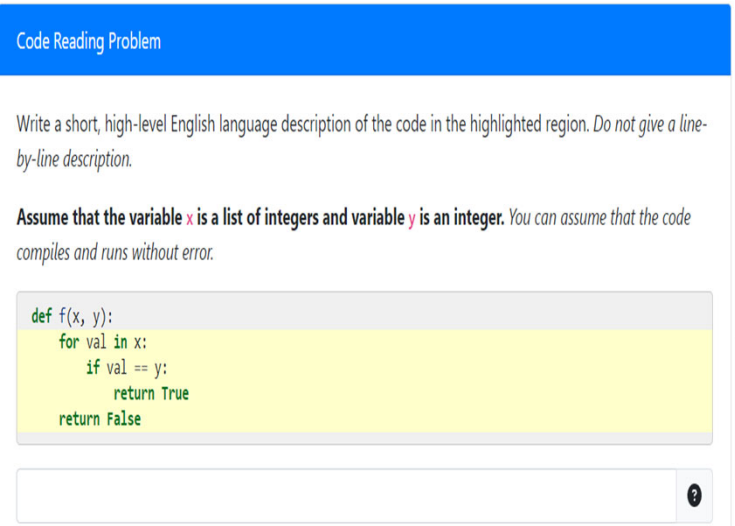
Explanation: provides a line-by-line description rather than a holistic description of the code's behavior or purpose

- 3 logistic regression models
- Correct vs Incorrect
- Unambiguous vs Ambiguous
- High-level vs low-level

Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

Comparing multiple autograders

Binary (Logistic, SVM)



Three "stacked" models – 3D

```
def f(x):
    z = []
    for y in range(x+1):
        z.append(y)
    return z
```

A desired answer: Return a list containing numbers from 0 to x
Ambiguous answer: Add the number between 0 and x to a list
Explanation: not clear that it is a new or existing list. Also, the singular "number" is confusing

```
def f(x):
    y = []
    for val in x:
        if val > 0:
            break
        y.append(val)
    return y
```

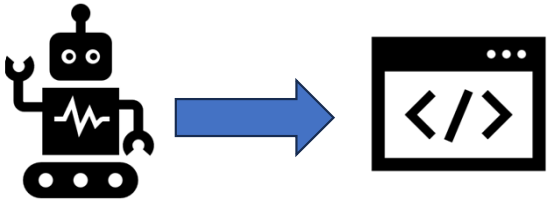
A desired answer: Makes a copy of a list up to the first positive number
Incorrect answer: Returns a list of only negative numbers from x
Explanation: indicates the wrong subset of the list

```
def f(x, y):
    if x < y:
        print(x, y)
    else:
        print(y, x)
```

A desired answer: Print two given numbers in numeric order
Low-level answer: Prints the y value then the x value if the x value is greater than or equal to y or prints the x value then the y value if x is smaller
Explanation: provides a line-by-line description rather than a holistic description of the code's behavior or purpose

Two LLM based graders

Returns whether a given list contains a given value



Returns whether a given list contains a given value

Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

GPT Prompt

Few shot "training"

5 correct answers

5 incorrect answers

Student answer

We later asked for feedback

Pretend that you are a teacher and you are teaching students Python. You ask students to describe in plain English what the following piece of code does with the assumption that the variable x is a list of integers and variables y and z are integers:

```
def f(x, y, z):
  for i in range(len(x)):
    if x[i] == y:
      x[i] = z
```

Example correct answers:
The function finds any appearance of the value of y in list x and replaces it with the value of z.
Finds all instances of an integer within a list, and replaces them with a different integer.
Reassigns specific numbers in a list with another number
changes the values of the integers in x which are equal to y into z
switches all numbers in a list that match a number (y) with another number (z)

Example incorrect answers:
gives a copy of a list if the number falls in the range
creates a new list the same length as orginal with duplicate values
replaces every element of variable y with z in list x
update the integer at index ith from y to z in the list x.
returns the value of a number if that value is found in the list

One student answered:
changes all instances of a certain number in a list to a different number

Is the student's answer complete, correct, unambiguous, and summarizing without details of implementations? (only *Yes.*/*No.*)

Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

GPT as a grader: example

This question is graded by orange otter 

Write a short, high-level English language description of the code in the highlighted region. *Do not give a line-by-line description.*

Assume that the variable x is a list of integers. *You can assume that the code compiles and runs without error.*

```
def f(x):
    y = 0
    for val in x:
        if val > 0:
            y += val
    return y
```

`Return the sum of all positive numbers from a given list`

Score: **100%**

I agree with GPT!

Message

```
Great job! The answer correctly and succinctly describes the purpose of the function without going into unnecessary details. Keep up the good work!
```

`Return the sum of all numbers from a given list`

Score: **0%**

I like this feedback

Message

```
Your answer is on the right path, but it missed one important detail. The code is indeed summing numbers from a list, but not all of them. Look more closely at the condition inside the loop to enhance your explanation.
```

`Return the sum of all positives from a given list`

Score: **100%**

Message

```
Great job! You understood the main purpose of the code correctly.
```

I might have marked this one wrong…

Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

Model performance so far

Grader	Accuracy (vs Ground Truth)
Bigram Logistic Regression (original)	86.3%

Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

Model performance so far

Grader	Accuracy (vs Ground Truth)
Bigram Logistic Regression (original)	86.3%
SBERT SVM	86.5%
OpenAI SVM	88.9%

Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

Model performance so far

Grader	Accuracy (vs Ground Truth)
Bigram Logistic Regression (original)	86.3%
SBERT SVM	86.5%
OpenAI SVM	88.9%
GPT-3.5 few shot	75.0%
GPT-4 few shot	86.6%

Roughly the same for less data

Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

Model performance so far

Grader	Accuracy (vs Ground Truth)
Bigram Logistic Regression (original)	86.3%
SBERT SVM	86.5%
OpenAI SVM	88.9%
GPT-3.5 few shot	75.0%
GPT-4 few shot	86.6%
GPT-4 Code Generation	82.7%
3D All Correct	85.5%

Possibly lenience?

Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

Small Pilot Study

- Pilot -182 students, 6 questions with each grader

Code Reading Problem for Lab

This question is graded by green guinea pig 

Write a short, high-level English language description of the code in the highlighted region. *Do not give a line-by-line description.*

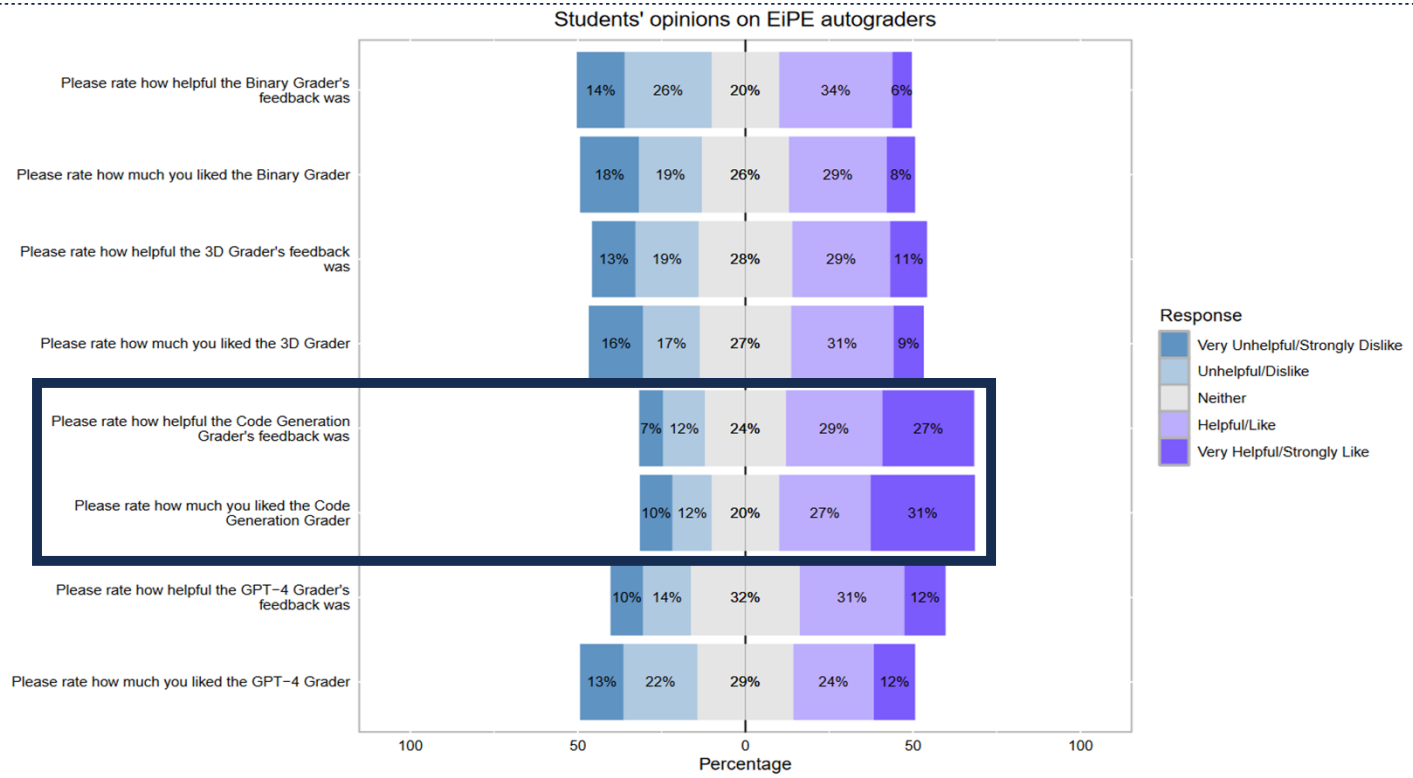
Assume that the variable x is an integer *You can assume that the code compiles and runs without error.*

```
def f(x):
    z = []
    for y in range(x+1):
        z.append(y)
    return z
```

Binary Grader	
3D Grader	
GPT-4 Grader	
Code Gen Grader	

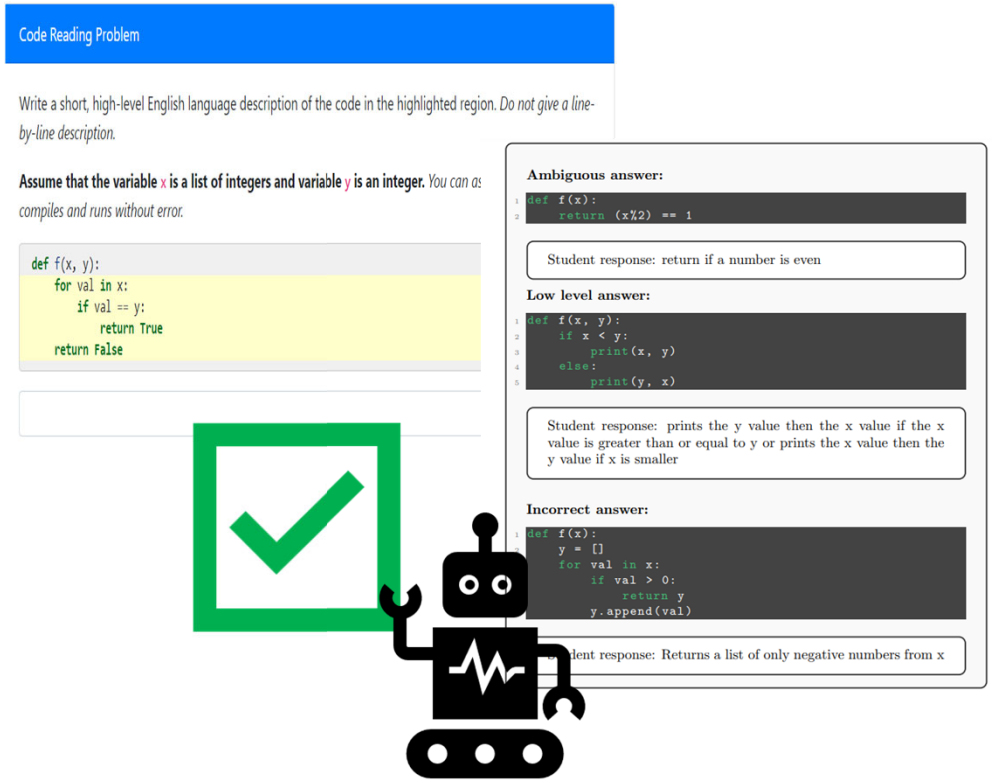
Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

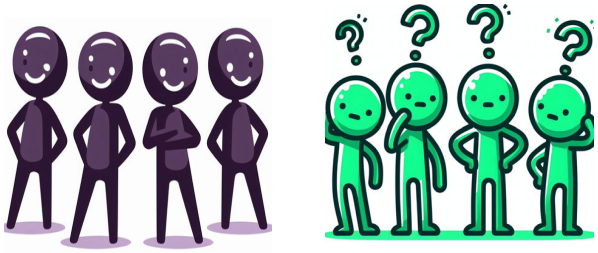
Students favored the Code Generation Grader

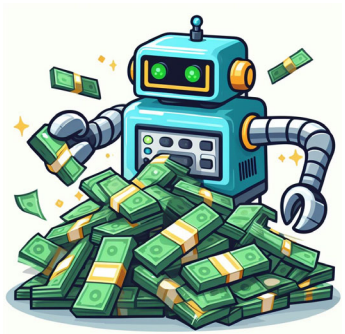


Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

Takeaways from model comparisons







Fowler et al. Evaluating AI models for Autograding Explain in Plain English Questions; Challenges and Considerations. In revision

Skills you may want to get involved!

- Useful interests/skills:
  - Analyzing data (in particular, score/exam data)
  - Python (used for question generating & research code)
  - Interest in education/learning theory (why do we want isomorphs)

Possibly useful classes

- CS 107/STAT 207 – for data analysis + Python (Ask your advisor)
- Ed Psych – EPSY 201
- CS 361/STAT 400 (Prob & Stats)
- CS 465? (UI)
- Ed stat, data science, (applied) ML can't hurt

I **just** focused on my assessment work

- I also care about broadening participation in computing -> CS+X curriculum project
- I also care about open education resources -> cleaning up CS 105 for broader sharing

Thank you!

Max Fowler
<mfowler5@illinois.edu> | mfwlr.github.io

My work uses **statistical and machine learning methods** and **qualitative methods** to determine how best to **conduct large scale, fair assessments** and **support instructors in the production and grading of questions** for such assessments.

Other projects to ask about: CS+X curriculum, open ed resources

The nicer graphics were generated using Bing's Dalle Image Generator -> prompts can be shared on request!