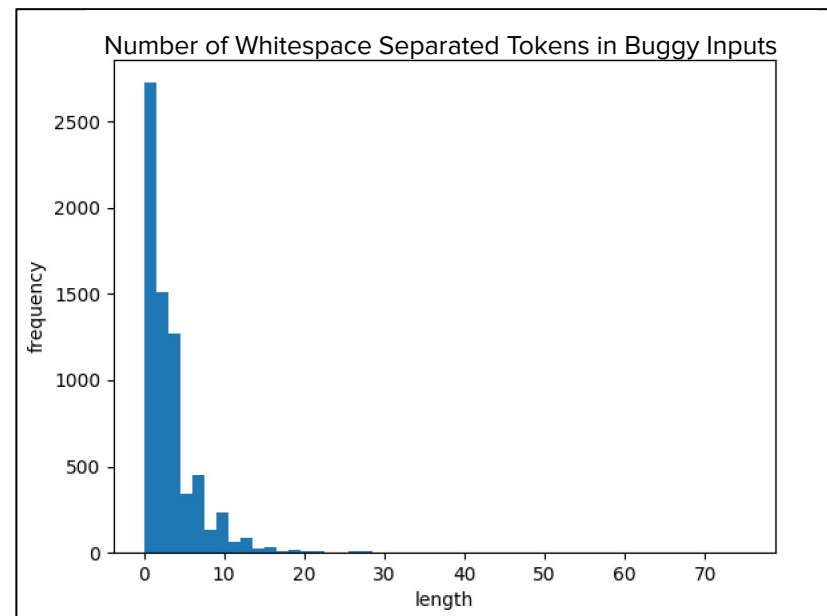


Automatically Repairing Input Data for Novice Python Programs

Madeline Endres, University of Michigan

Why Input-Related Bugs

- Access to 4 years of Python Tutor data thanks to Philip Guo
- 35% of python programs contain a call to `input()`
- Found **over 25,000 buggy input / program pairs** where only the input differed in the student's "fixed" version



Example Input-Related Error

In practice, some error messages novices face are fixed by only changing the program's *input*:

Example of Simple Syntactic Mistakes:

Code:

```
x = float(input())  
print(x * math.e / 2)
```

Error Causing Input:

5,2

Student's Fix:

3.1

Error = Python expects period decimal notation:

```
ValueError: could not convert  
string to float: '5,2'
```

More Complex Buggy Input Data Example

Buggy Input:

```
abcd
*d%#
abacabadaba
#*%*d*%
```

Error:

```
Traceback (most recent call last):
line 13, in <module>
rashifr_itog += slovar[rashifr[k]]
KeyError: '#'
```

```
1  ishodniy = input()
2  konechniy = input()
3  zahiffr = input()
4  rashiffr = input()
5  zahiffr_itog = []
6  rashiffr_itog = []
7  slovar = {}
8  for i in range(ishodniy.__len__()):
9      slovar[ishodniy[i]] = konechniy[i]
10 for j in range(zahiffr.__len__()):
11     zahiffr_itog += slovar[zahiffr[j]]
12 for k in range(rashiffr.__len__()):
13     rashiffr_itog += slovar[rashiffr[k]]
14 print(zahiffr_itog)
15 print(rashiffr_itog)
```



Observations about Input-Related Interpreter Errors

- For syntactic errors, the **error message** is highly correlated to the **eventual student fix**
- For complex errors, fixes are **more diverse**, but we observed that some fix mutations were more common than others. E.g.:
 - Inserting a string literal from the program
 - Inserting a small integer
 - Swapping two lines of inputs
 - Splitting an input line on whitespace
- Student repairs are generative, not just corrective
 - Often requires multiple error messages to be fixed before finding solution

InFix Algorithm

- Iterative search-based algorithm that modifies the student's error-causing input.
- Use **error message templates** to try and repair common syntactic errors
- Apply a random **additional mutations** for non-templated error-messages

Example of Algorithm Fix

Python Program

```
1 def main():
2     m=int(input('inserire un intero '))
3     L=list(input('inserire stringhe '))
4     s=''
5     s=concatena(m,L)
6     print(s)
7
8
9
10 def concatena(m,L):
11     if m!=type(int)or L!=type(str):
12         print('None')
13 main()
```

Original Bad Input:

`ciao`

Iteration 1 = ValueError template:

`-1`

Iteration 2 = Mutation template :

`-1`

`ciao`

Evaluation Results

- Empirical results: Can **fix 95%** of 25,000 input-related errors
- Human Study results: 97 participants found the machine repairs of **equal helpfulness** and within **4% the quality** to student made repairs

Questions?