

Analyzing Software using Deep Learning

Introduction of Course Project

Prof. Dr. Michael Pradel

Software Lab, University of Stuttgart

Summer 2022

Overview

- **Task**
- **Scripts and Resources**
- **Organization**

Motivation

- **Exceptions should be**
 - Raised only when some **error condition** occurs
 - Otherwise: Unexpected crash
 - **Convey the reason** for being raised
 - Hard to debug

Examples

```
if mode not in {'caffe', 'tf', 'torch'}:  
    raise ValueError('Expected mode to be  
        one of `caffe`, `tf` or `torch`.  
        Received: mode={mode}')
```

Examples

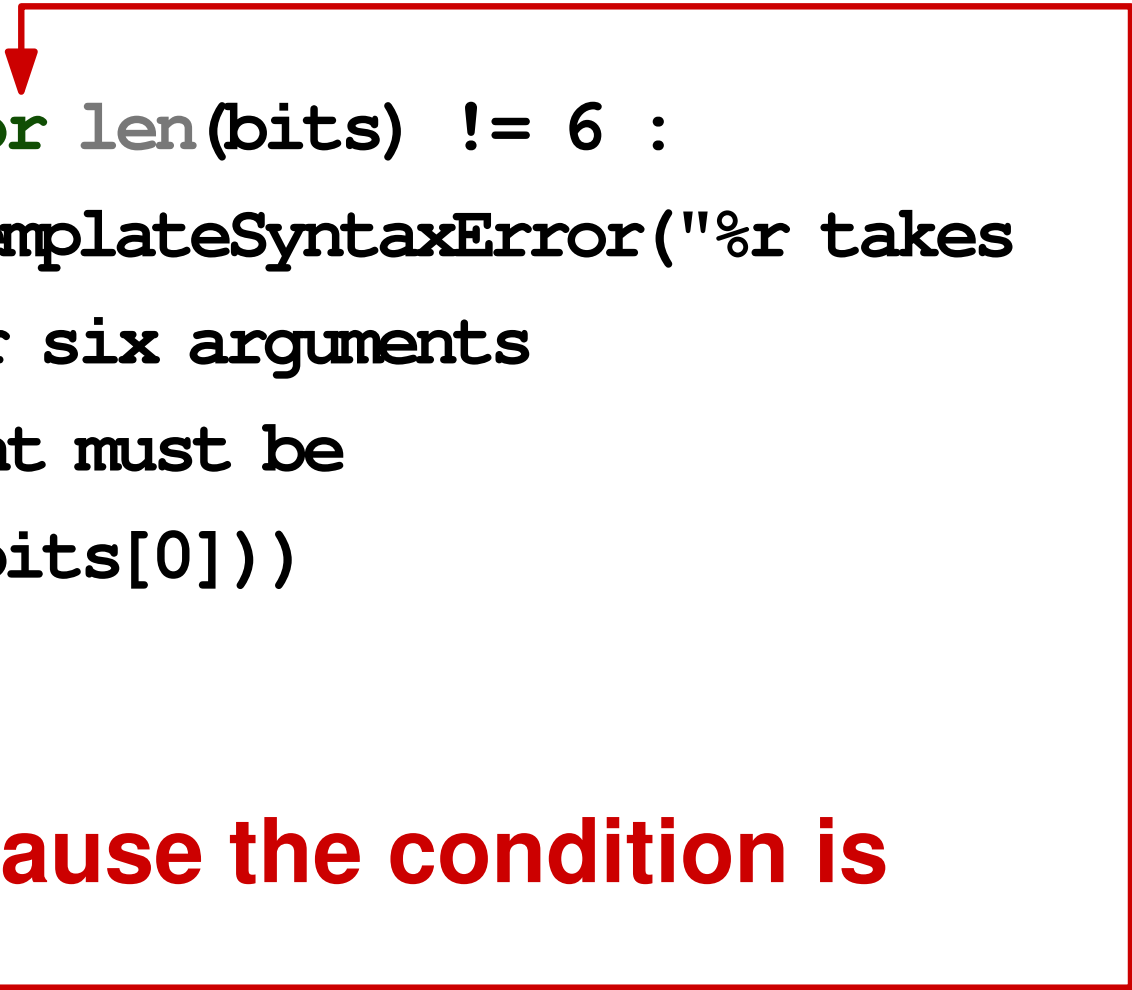
```
if mode not in {'caffe', 'tf', 'torch'}:  
    raise ValueError('Expected mode to be  
        one of `caffe`, `tf` or `torch`.  
        Received: mode={mode}')
```

Condition and exception are consistent

Examples

```
if len(bits) != 4 or len(bits) != 6 :  
    raise template.TemplateSyntaxError("%r takes  
    exactly four or six arguments  
    (second argument must be  
    'as' )" % str(bits[0]))
```

Examples



```
if len(bits) != 4 or len(bits) != 6 :  
    raise template.TemplateSyntaxError("%r takes  
        exactly four or six arguments  
        (second argument must be  
        'as' )" % str(bits[0]))
```

Inconsistent because the condition is wrong

Examples

```
if n2 > n1 :  
    raise ValueError('Total internal reflection  
    impossible for n1 > n2')
```


Examples

```
if n2 > n1 :  
    raise ValueError('Total internal reflection  
    impossible for n1 > n2')
```

**Inconsistent because the error message
is wrong**



Goal

Design, implement, and evaluate a **neural network-based program analysis that **identifies inconsistent if-condition-raise statements****

- Target language: Python
- Extract all if-condition-raise statements
- Predict probability that inconsistent (i.e., binary classification)

Dataset

- **100k Python functions**
 - Each with one or more if-condition-raise statement
 - Use for training and validation
- **10 consistent and 10 inconsistent if-condition-else statements**
 - Use for testing
 - For grading, we have more like this

Token Embedding Models

Two pre-trained token embedding models

- FastText model trained on 6.5M if-condition-else statements
- BPE tokenizer trained on 1.5M functions

(In)consistent Examples

- **Training a binary classifier:**
Need **both consistent and inconsistent** examples
- **Consistent** examples
 - Assume all code in the given 100k functions to be correct
- **Inconsistent** examples
 - Create them by **recombining and mutating the consistent examples**

Parsing and Manipulating Code

- **Parse** functions into ASTs
- Extract **raise statements** and **conditions** guarding them
- **For creating inconsistent examples**
 - Modify and recombine (sub)trees
 - Print trees to code again

Libraries to Use

- For parsing and code manipulation:
LibCST
- For deep learning:
PyTorch

Organization

- Each student has a **mentor**
- Three **milestones**
 - Meetings with mentor after each milestone:
May 30/31, June 13/14, June 27/28
- Project **submission deadline: July 15**
- Oral **presentation: July 18/19**

Milestone 1

- **Extract** if-condition-raise statements from the provided dataset
- Recommended: **AST-based** extractor built with LibCST

Milestone 2

- **Binary classifier**

- Input: If-condition-raise statement
- Output: Probability that inconsistent

- **Based on a simple approach to create inconsistent examples:**

Random recombination

Milestone 2

```
# statement 1: consistent
if x < 0:
    raise ValueError("x should be positive,
        got x={v}".format(v=x) )

# statement 2: consistent
if not isinstance(result, dict):
    raise TypeError("expected result to be of type dict")

## recombination 1: likely inconsistent
if x < 0:
    raise TypeError("expected result to be of type dict")

## recombination 2: likely inconsistent
raise ValueError("x should be positive,
    got x={v}".format(v=x) )
```

Milestone 3

- **Binary classifier that is effective for real-world examples**
- **Improvements over Milestone 2**
 - Better strategies for generating inconsistent examples
 - Improved neural model

Rules

- Preserve the **interfaces** of the provided **main scripts**
- Project is **individual**: No sharing of solutions
- Use **only the provided dataset**