

Assignment #1

Generating and testing code

Dr. Daniel M. Yellin

DRAFT

THESE SLIDES ARE THE PROPERTY OF DANIEL YELLIN
THEY ARE ONLY FOR USE BY STUDENTS OF THE CLASS
THERE IS NO PERMISSION TO DISTRIBUTE OR POST
THESE SLIDES TO OTHERS

Due date

Assignment is due May 1, 2024 at 17:00

17:00 @ 01.05.2024

Generating and testing code samples using different parameters

The goal of this assignment is to build a *program* that

1. generates different code samples using different prompts and different temperatures,
2. tests each code sample with unit tests,
3. and generates a file with statistics stating how many samples passed all unit tests per temperature and prompt.

This is all done programmatically, without any human intervention.

Input

Note: you no longer have to retrieve input of file name. It is always “input.txt”.

The input to your program will be a file named “**input.txt**” that contains:

1. **prompt**: A (textual) **description** the function to be created. It also contains the **function signature** (name and parameters).
2. **examples**: 1-3 example inputs and outputs.
3. **temperatures**: Two temperatures, where each temperature is a real number between 0-1
4. **k**: An integer k , $1 \leq k \leq 5$, the number of code samples to generate for each temperature.
5. **unit_tests**: A set of unit tests

Format of input and example

<function_name>

"""

signature: <signature>;

<description>

"""

example: <example 1>;

example: <example 2>;

temperatures: t1, t2;

num = k;

unit tests:

<unit test 1>;

<unit test 2>;

sqrt_list

"""

signature: sqrt_list(l:list) -> float;

A function named sqrt_list that takes a list of integer and returns the square root of the sum of that list, to two decimal places.

"""

example: sqrt_list([1,7,8]) -> 4.0;

temperatures: 0, .8;

num = 3;

unit_tests:

sqrt_list([6,18,7,5]) -> 6.0;

sqrt_list([50,2,33,8,4,3,10]) -> 10.48;

This is a unit test

This is the
“ground truth” –
the value that
should be
returned

Format of input (continued)

1. There may be 1 or many examples
2. There may be 1 or many unit tests
3. The input may contain additional whitespace and additional newlines and these are not significant.
4. We call everything in the Python docstring the **basic prompt**. (Everything inside *triple (3)* quotation marks is a docstring in Python).

The program you write should automate all steps of the process.

Program flow, part I

The program flow should work as follows:

1. Open the file “input.txt” and read the file. Read the **basic prompt**.
2. Call an LLM via its API using the **basic prompt** to generate k samples for each temperature.
3. Read the examples from input.txt
4. Append the examples to the basic prompt creating a new prompt and repeat steps 1-3 with this new prompt

Program flow , part II

5. Make sure the sample code has no leading or trailing text. You may need to remove leading and trailing text from the program generated by the LLM.
 - You can do so either by checking programmatically that there is no extra text, or by first running the program and seeing it fails for this reason. You can then remove the extra text programmatically or ask the LLM to clean it up.
6. Add the unit tests to invoke the generated function (code sample), forming a program P (P = generated function + unit tests).
7. Write P to a file in the current directory. See next slide for how to name the file.
 - Note you will have $k \times 2 \times 2$ different code samples = $k_samples_per_temperature \times 2_temperatures \times 2_prompts_per_temperature$

Naming the code samples generated

<name> is the name of the function (i.e., <function_name>)

<sample#> is an integer between 1 and k, the sample generated by the LLM for a specific prompt

<temp> is the temperature used to generate the sample without the decimal point “.”. Specifically:

- if temperature = 0, <temp> = 0
- if temperature = .7, <temp> = 7
- if temperature = 1, <temp> = 10

Name each generated code sample where no examples were given as follows:

<name>-s<sample#>-t<temp>.py

Example: [max-s2-t6.py](#)

Name each generated code sample where examples were given as follows:

<name>-ex-<sample#>-<temp>.py

Example: [diffset-ex-s1-t9.py](#)

Program flow, part III

8. Run the program P and check the output of the unit tests against the “ground truth”
9. For each program P, for each code sample, write the following to a file **<name>-stats.txt**

LLM=<name_of_LLM>
temperature= t1
passed_without_examples=x
failed _without_examples= y
average _without_examples = $x/(x + y)$

LLM=GPT3.5-turbo
temperature=.8
passed_without_examples=1
failed _without_examples= 2
average _without_examples=.33

passed_with_examples=x
failed _with_examples= y
average _with_examples = $x/(x + y)$

passed_with_examples=3
failed _with_examples= 3
average _with_examples=1.0

10. Also record the same information in the **same** file for the second temperature

Testing

We will test your code by running it with a file name for input.

We will see that:

1. It generates all the code samples required, following the instructions in the input file
2. It correctly tests each code sample
3. It writes the summary statistics to a file as specified

Submit assignments using co-lab

Use Google Colab, <https://colab.research.google.com/>

After you complete your assignment and test it, follow the instructions given in Moodle Assignment1.

- Save your python notebook to your computer with the name *gencode.ipynb* extension
- Your notebook can invoke other python (.py) files
- Zip the notebook (*gencode.ipynb*) and any other files that your notebook requires
- The Zip file should be named *hw1_<student1TZ>_<student2TZ>.zip*, where you give the תעודת זהות of each student in the group. For example, *hw1_123456789_356789544.zip*
- Upload it to Moodle

The בודק is Yam Libman.

The LLM API key

- You can use any LLM that you want, but you have to have it in your program so that the בודק can check the assignment.
 - After he checks your assignment, you can delete that key
- I am working on a solution so you can use a class account and API key for Azure OpenAI. If so, the בודק will not need your key. I have not yet closed on this solution.