CSE 411: ADVANCED PROGRAMMING TECHNIQUES

Fall 2016

Programming Assignment 3: GUI Programming Assignment

Sachin Joshi

Homework Description

The assignment consists of a GUI program in Python, "**saj415_hw3.py**", that uses the Tkinter library to present a graphical interface that allows users to experiment with the edit distance algorithm. The interface takes the following inputs from the users:

- Positive integer values for **cins**, **cdel**, **csub**.
- Input string S and input string T.

The program computes the edit distance between the two strings entered by the user, using the indicated cost assignments and displays the following information:

- The full edit distance matrix, if the associated button is checked.
- The backtrack matrix, if the associated button is checked.
- The alignment, if the associated button is checked.

Running the Program

The program can be directly executed by double clicking the **saj415_hw3.py** Python file. We can also run our program from the command prompt using the following syntax:

python saj415_hw3.py

Edit Distance GUI

- Enter filenames in the source and the target text area.
- Set the cost of insertion, deletion and substitution according to your requirements.
- Select the appropriate checkbox for the kind of output you require.
- Click on the **Compute** button for displaying output in the text area.
- Click on the **Clear** button to clear the output text area.

Test Files

- **testfile1.txt:** Contains the string "**compare**".
- **testfile2.txt:** Contains the string "**computer**".

Method Description

1. **__init__ method**: Initializes the Application Frame for the program.
2. **create_widgets**: Method to create different widgets required for this particular program GUI.
3. **initialize**: Method bound to the Compute button and is called when it is selected. It handles the calling of all the important functions of this program.
4. **clearOutputWindow**: Method to clear the output text area.
5. **findEditDistance**: Method to create the edit distance matrix and the backtracking matrix for two strings present in the filenames mentioned in the source and the target text area.
6. **getMinimumDistance**: This method returns the most minimum value amongst the 3 operations performed on a matrix element - insertion, deletion and substitution.
7. **getStringAlignment**: This method returns the aligned string for the source and target strings.
8. **getFileString**: This method reads the string from the filename mentioned in the text area and returns the entered string. It also handles different error scenarios.
9. **convertMatrix**: Method that converts the matrix into a printable string format on the result window.
10. **convertEditDistanceMatrix**: This method does the same thing as convertMatrix method but is specialized to display the edit distance matrix.

Note

- It is mandatory that the strings to be compared are present in the test files that are being mentioned in the input text area.
- It is mandatory that the string in the file is present in the first line, otherwise the program will not execute.
- The test files need to be present in the same file structure as saj415_hw3.py.