**Homework 2: Edit Distance**

**Aim:**

Create a simple spell checker program using python.

**Observation Criteria and Analysis:**

Dataset: Dictionary created with dataset derived from “War of Worlds” as a part of Homework 1.

1. Edit Distance Algorithm:
   1. As part of the algorithm, length of input word (from string passed by user) and the length of dictionary word is calculated.
   2. Created a 2-D matrix (distance matrix) and initialized the matrix with (0,0).

Approach 1: initially, considered the ranges/distance in the matrix as the length of the input word (n) and length of the dictionary text (m) which resulted into an ‘Error: list index out of range’, implying that the elements being accessed don’t exist in the list.

Approach 2: Moved on to change the ranges in the matrix as the length of the input word + 1 (i.e., n+1) and length of the dictionary text+1 (i.e., m+1). This rectified the error mentioned above.

* 1. Initialized the zeroth row and column is the distance from the empty string, after which we initialized the first loop which iterates over n values, assigning ‘i’ to the initial member of sub-array, meanwhile the second loop, loops over m values, setting the first element of each sub-array of the first dimension to ‘j’.
  2. Implementing the recurrence relation of the edit distance algorithm which implements the following:

For each i = 1....n

For each j = 1….m

D(i,j) = min [D( i-1, j) + 1, D( i, j-1) + 1, D(i-1, j-1)+ d]

d -> 2, if X(i) != Y(j)

d -> 0, if X(i) = Y(j)

* 1. Finally, store the edit distance into a variable. This distance will be further used to calculate the closest word.

1. Spell checker function
   1. Pass the dictionary.txt file, which contains all the words that were normalized and processed as a part of homework 1.
   2. User input text is taken and normalization process such as lowering upper case characters to lower case characters, replacing special characters, numbers and punctions and tokenizing the string is performed.
   3. Using looping, minimum distance and closest word for each token (words that were part of the sentence passed by the user).

Approach 1:

* Initially, declared the minimum distance to positive infinity, after which we declare a variable to store the closest word.
* Using a for loop, call the edit distance algorithm passing user passes text and words from created dictionary and store the value of the distance.
* Further to this, distance calculated earlier is matched with minimum distance, and if the value if less, assign the calculated value to the minimum distance and word from the dictionary to the closest word.

Problem with the approach:

* The loop wasn’t iterating through all the words in the dictionary, to find the closest word.
* Return value for every input was the same.

Approach 2: (adding on to the previous)

* In this approach, tokens (individual words from user string) are looped through using a for loop.
* For token not existing in the dictionary, initialize the minimum distance to positive infinity and another for loop is called to loop through the dictionary and calculate the edit distance.
* In addition to this, distance calculated earlier is matched with minimum distance, and if the value if less, assign the calculated value to the minimum distance and word from the dictionary to the closest word.

Problems with the approach:

* For a sentence, for each loop the closest word was overwritten with new suggestion.

Approach 3: (adding on to the previous approach)

* Rather than declaring the closest word variable as a string, closest word is declared as a list, and after every iteration the closest word is appended to the list created.
  1. Additionally, conditions such as input “quit” simply exits the program and if the tokens (input from the user string) is present in the dictionary, the code returns no misspells detected.

**Summary:**

1. Edit distance algorithm was successfully implemented to calculate the minimum distance between words and find out the most closest word to the misspelled word using the dataset provided in our dictionary.
2. However, since the dictionary is limited with words given in the book ‘War of Worlds’ closest words suggested isn’t always accurate.