

**Name:** Sanket S. Fulzele

**Roll No:** 371018

**PRN:** 22110728

**Div:** A

✓ 1. Introduction of Available Word Similarity Measures:

- Provide an overview of word similarity measures commonly used in Natural Language Processing (NLP).
- Discuss the importance of word similarity measures in tasks like spell checking, auto-correction, and information retrieval.
- Explain the concept of Minimum Edit Distance (also known as Levenshtein distance) and its significance in measuring the similarity between two words.

✓ 2. Manual Calculation of Minimum Edit Distance:

- (Saturday, Sunday): 3
- (Eaten, Beaten): 2
- (Minimum, Maximum): 3

✓ 3. Create a List of Lexicons for the Spelling Checker:

- Compile a list of lexicons or dictionaries for two languages of your choice.

- Ensure the lexicons cover a wide range of vocabulary and include commonly used words in the respective languages.
- Each entry in the lexicons should include the word's correct spelling and any additional information (e.g., part of speech, frequency of use).

#### ✓ 4. Develop a Spelling Checker using Minimum Edit Distance:

- Implement a spelling checker using the Minimum Edit Distance algorithm for the two chosen languages.
- The spelling checker should take input text and identify misspelled words by comparing them to the words in the lexicons.
- Provide suggestions for correcting misspelled words based on the words in the lexicons that have the shortest edit distance.
- Ensure the spelling checker is user-friendly and capable of handling input text efficiently.

#### ✓ 5. Conclusion:

- Summarize the key findings and outcomes of developing the spelling checker using Minimum Edit Distance.
- Reflect on the challenges encountered during the development process and discuss any insights gained.
- Offer recommendations for improving the spelling checker or extending its functionality in future iterations.

```

1  import nltk
2
3  # Download NLTK word list (if not already downloaded)
4  nltk.download('words')
5
6  from nltk.corpus import words
7
8  def min_edit_distance(source, target):
9      m = len(source)
10     n = len(target)
11
12     # Initialize a matrix to store the edit distances
13     dp = [[0] * (n + 1) for _ in range(m + 1)]
14
15     # Initialize the first row and column
16     for i in range(m + 1):
17         dp[i][0] = i
18     for j in range(n + 1):
19         dp[0][j] = j
20
21     # Fill in the matrix using dynamic programming
22     for i in range(1, m + 1):
23         for j in range(1, n + 1):
24             cost = 0 if source[i - 1] == target[j - 1] else 1
25             dp[i][j] = min(dp[i - 1][j] + 1,          # Deletion
26                           dp[i][j - 1] + 1,          # Insertion
27                           dp[i - 1][j - 1] + cost)    # Substitution
28
29     return dp[m][n]
30
31 def spelling_checker(word, dictionary):
32     # Find the closest match in the dictionary
33     min_distance = float('inf')
34     closest_match = None
35
36     for candidate in dictionary:
37         distance = min_edit_distance(word, candidate)
38         if distance < min_distance:
39             min_distance = distance
40             closest_match = candidate
41
42     return closest_match, min_distance
43
44 # Take user input
45 word_to_check = input("Enter a word: ")
46
47 # Use NLTK words list as the dictionary
48 dictionary = words.words()
49
50 closest_word, min_distance = spelling_checker(word_to_check, dictionary)
51 print(f"Suggested correction for '{word_to_check}': {closest_word}")
52 print(f"Minimum edit distance: {min_distance}")

```

# OUTPUT:

```
[nltk_data] Downloading package words to
[nltk_data]      C:\Users\Dell\AppData\Roaming\nltk_data...
[nltk_data]   Package words is already up-to-date!
Enter a word: helllo
Suggested correction for 'helllo': hello
Minimum edit distance: 1

Enter a word: summeer
Suggested correction for 'summeer': summer
Minimum edit distance: 2

Enter a word: eeneergy
Suggested correction for 'eeneergy': energy
Minimum edit distance: 2

Enter a word: Happinneess
Suggested correction for 'Happinneess': happiness
Minimum edit distance: 3

Enter a word: intelligence
Suggested correction for 'intelligence': intelligence
Minimum edit distance: 0
```