Documentation Details for Capstone Project

P8 - Spell Checker
TEAM - BUG WASHERS

MEMBERS - 1) Jaimin Hadvani (202301146)
2) Het Ladani (202301102)
3) Utsker Dhameliya (202301083)
4) Pal Kaneria (202301021)

In this project, we are implementing a spell checker program in C++. Algorithm for the code is given below. This program aims to identify and correct the misspelled words in a given text document.

Steps or parts of code(Algorithm):

1) Trie Data Structure:
   - ❖ The program uses trie data structure to store the dictionary of words. Each node in the trie represents a character in a word, and edges represent the transition from one character to another. The Node class represents each node in the trie, containing a map of child nodes for each character and a boolean flag indicating whether the node represents the end of a valid word.

   - ❖ The map data structure is used in the Node class to store child nodes, where the key represents a character and the value represents the corresponding child node.
   - ❖ Vectors are used to store the suggested spellings

2) Distance Calculation:

   In this code, we are calculating the edit distance between two strings. The computeEditDistance function computes the minimum number of operations (insertion, deletion, substitution) required to transform one string into another.

3) Spell Checking and Suggestions:

   The SpellChecker class offers functionalities to manage a dictionary, search for words, and obtain suggestions for misspelled words.

- The findSuggestions function iterates through the trie recursively to locate words resembling the misspelled word, considering their edit distance.
- Suggestions are gathered in a vector and sorted according to how closely they match the misspelled word based on edit distance.

4) Highlighting Misspelled Words:

   If the user wants to see how many misspelled words in their provided text document, our code does the work of highlighting those misspelled words by showing them in red colour.

5) Text Correction:

   The program can read a text document, identify misspelled words, and provide suggestions for corrections.

   The code will first display the misspelled words and highlight them and with the user's approval, the misspelled words will be replaced by the correct spellings.

Additional features of our code

1) **Command Line Interface:**
   The program provides a command-line interface for users to interact with the spell checker

2) **Automatic Correction:**
   The program offers an option for automatic correction, where it automatically selects the closest match from suggestions.

3) **Word Count:**
   Users can count the total number of words in the input text document.

Why we chose trie over other data structures?

Trie is a tree-based data structure used for efficiently storing and searching for strings in a set of data.

- One of the advantages of using trie over other data structures is its property of fast searching operation i.e we can search for a key by traversing down the tree from the root and the time of searching is directly proportional to the

length of the key which makes the data structure efficient to search keys in large dataset.

- Secondly, it is space efficient because they store only the characters that are present in the keys, and not the entire key itself. Hence the space complexity of the trie depends on the number of nodes present in a trie.

Moreover, deletion is easier with tries where O(l) is its time complexity, and l is the length of the word to be deleted.

Drawbacks of other data structures-

HASH MAPS:

- ➢ **Space Complexity:** Hash maps might use more memory than tries, especially with sparse data or long common prefixes in keys. This is because hash maps need additional memory for storing hash values. In the average case, space complexity is O(n), where 'n' is the number of elements. However, in the worst-case scenario, space complexity can come down to O(n^2).
- ➢ **Time Complexity:** Although hash maps generally provide constant-time average-case performance (O(1)) for insertion, deletion, and lookup operations, their worst-case time complexity can degrade to O(n) due to collisions, where n is the number of elements in the hash map.

LINKED LIST:

- ➢ **Space Complexity:** Linked lists have a higher space complexity compared to tries due to the need to store pointers for each node, resulting in increased memory usage. The space complexity of a linked list is O(n), where 'n' is the number of elements in the list.
- ➢ **Time Complexity:** While linked lists offer O(1) time complexity for insertion and deletion at the beginning or end, trie operations typically boast more efficient time complexities, particularly for string-related tasks.

STACK:

- ➢ **Space Complexity:** Stacks can use more memory than tries, especially for big data. Stacks allocate memory for each added item. In the worst-case scenario, the space complexity of a stack would be O(n), where 'n' is the number of elements in the stack.
- ➢ **Time Complexity:** Stacks are fast for basic operations like adding and removing items (O(1)), but they're not great for string tasks like finding prefixes or subwords.

QUEUE:

- ➢ **Space Complexity:** Queues typically allocate memory for each element enqueued, leading to potential memory overhead. Its space complexity is O(n), where 'n' is the number of elements.
- ➢ **Time Complexity:** While queues provide efficient time complexity for basic operations like enqueue and dequeue (O(1)), they may not be suitable for tasks involving string-related operations, such as prefix matching or substring search.

Time Complexity:

Addword(for adding a word in a dictionary), Removeword(for removing a word from the dictionary), SearchWord(for searching a word in a dictionary).
The time complexity of these functions is as below:

1) Adding a word - O(k), where k is a length of string which is constant,hence it almost takes constant time for adding a word in a dictionary.
2) Removing a word - O(k), where k is again length of string which is constant, hence removing a word from dictionary also happens in almost constant time.
3) Suggesting a word - O(n) , n is number of strings . Suggesting a word takes more time than adding or removing a word in a dictionary.

PSEUDOCODE -:

1. Begin

2. Create a class node for making a map.

   - ➢ In class bool endofword checks that the word has ended or not

3. Create a spellchecker class that contain a private member called root node which is a pointer of the type node class

   - ➢ Make a function methodcomputeeditdistance which will return a number of character difference between given two strings.
   - ➢ Define a function for finding suggestions for the misspelled word.
   - ➢ Create a addword function which adds the given word in the dictionary (if the user wants)
   - ➢ Create a removeword function to remove a word from the dictionary (if user wants)
   - ➢ Define a searchword function to search a word in the dictionary

- ➢ Create a getwordsuggestion to give suggestion for similar words.
- ➢ Define a function removespecialcharactar to removes the special characters like full stop (.) and coma(,)
- ➢ Define a split to store a suggestion for individual word in a vector

4. In main function

- ➢ if dictionary is not found then giving an error or if open then store all the word in trie
- ➢ ask a user for enter a file name
- ➢ if provided file exists then go to other task else print error
- ➢ ask a user for the operation they wanted to execute
- ➢ close the file

5. end

OUTPUT -:

```
PS C:\Users\Dell> cd "c:\Users\Dell\Desktop\coding\c++\sets\capstone\" ; if ($?) { g++ spellchecker
4.cpp -o spellchecker4 } ; if ($?) { .\spellchecker4 }
please enter your text file name here :text.txt

*****************************************************************************************************
**********************************

******-----> enter 0 to exit the loop

******-----> enter 1 to add a word in dictionary

******-----> enter 2 to remove a word in dictionary

******-----> enter 3 to highlight a Misspelled word

******-----> enter 4 for selecting the appropriate suggestion to replacing the misspelled word with
 the correct one

******-----> enter 5 to automatically update the paragraph

******-----> enter 6 to word count

********** Enter the number corresponding to your need **********

*****************************************************************************************************
**********************************
```

```
The man walked down the  streat enjoying the sunshine and the warm breeze He stopped to admire the  beutif
ul flowers blooming in the  gardan Suddenly he  realised he had forgotten to  tak an important call earlie
r that  dey
*****************************************************************************************************
****************************
```

```
 Suggestions for 'streat':

1.   streak
2.   stream
3.   street
4.   treat

Enter the number corresponding to the correct spelling or 0 to keep the original word:
3

 Suggestions for 'beutiful':

1.   beautiful

Enter the number corresponding to the correct spelling or 0 to keep the original word:
1

 Suggestions for 'gardan':

1.   garden

Enter the number corresponding to the correct spelling or 0 to keep the original word:
1
```

```
 Suggestions for 'realised':

1.   realise
2.   realized

Enter the number corresponding to the correct spelling or 0 to keep the original word:
2

 Suggestions for 'tak':

1.   tab
2.   tad
3.   tag
4.   take
5.   talk
6.   tan
7.   tank
8.   tap
9.   tar
10.   task
11.   taw
12.   tax
13.   teak
14.   yak

Enter the number corresponding to the correct spelling or 0 to keep the original word:
4
```

```
 Suggestions for 'dey':

1.   day
2.   defy
3.   den
4.   deny
5.   dewy
6.   dry
7.   key
8.   ley

Enter the number corresponding to the correct spelling or 0 to keep the original word:
1

*****Text has been corrected and saved to 'corrected_text.txt'.*****
```

CORRECTED TEXT-:

corrected_text.txt

```
1   The man walked down the street enjoying the sunshine
2   and the warm breeze. He stopped to admire the beautiful
3   flowers blooming in the garden Suddenly, he realized
4   he had forgotten to take an important call earlier
5   that day.
```

ADDITIONAL FEATURE OF OUR CODE -:

1) *CUSTOM DICTIONARY*

```
1

How many word you want to add in dictionary :
2
provide
offer
Word added successfully in dictionary
```

```
2

enter the word that you want to remove
provide
Word successfully deleted
```

2) *AUTO-CORRECTION*

```
5

 WARNING : you may or may not get corrected spelling of your words

Text has been corrected and saved to 'corrected_output.txt'. Thank you
for using our spelling correction program.
```

Output of auto-correction -:

corrected_output.txt

```
1   The man walked down the streak enjoying the sunshine
2   and the warm breeze. He stopped to admire the beautiful
3   flowers blooming in the garden Suddenly, he realise
4   he had forgotten to tab an important call earlier
5   that day.
```

3) *WORD COUNT*

```
6

Word Count : 38
```