Michael Ekubay Negassi (002238528) & Lu Yang (002238352)

1. Introduction

This Project is part of the summer project in completion of CS590. The project is about Spelling Checker and it demonstrates the basic spelling problems and their algorithmic solution, we use Java with Swing GUI for implementation and Bloom Filter as an algorithm to provide a reliable solution for spelling mistakes.

2. Prerequisite

It requires **JDK** and **JRE** version 10.0.2 or higher, **SWING** eclipse GUI plugin for Eclipse Oxygen Designer, the provided Corpus of text and created a dictionary from it. Once the dictionary is created we can take input from a text file and correct spelling mistakes if any.

3. Approach Bloom filter algorithm

Bloom filter algorithm is a data structure algorithm. which is fast and efficient in determining whether an element is a member of a set or not. It is mostly used in applications such as spell checkers, differential file updating, distributed network caches, and textual analysis. We use MD5 which is famous cryptographic hash function that maps arbitrary sized data input into 128-bit hash value.

Bloom filter is a data structure utilizes hash functions to support its two core operations; element addition and element existence checking. The algorithm checks a given element and returns false if the element is not in the set and true if the element is probably in the set. Most of the time elements can be added to the set but not removed from the set therefore the two basic operations of this algorithm are elements addition and elements existence checking.

For the algorithm to perform those operations it uses any number hash functions and arrays. The use of hash functions in a data structure is to map the data input to the array when adding or checking an element. The hash function is useful because, it quickly locates the data records by an index key. The array is used to store the input data.

The Bloom filter algorithm uses bit array of m bits as its data structure to store set of elements. Moreover, it has different k hash functions which are used to map set of elements to the m bit array. When storing elements in the m bit of array it generates a number, which help to determine the positions of the array and mark them as occupied. At first, the bit array is initialized with value 0 on all its elements. The value 0 determine that no element is added in the bit array.

Then, the first operation which is elements addition operation is done by inserting the element to each of k hash functions to get k array positions. These k bits are all set to 1. Similarly, other elements are also added in a same fashion. Meanwhile the second operation which is the element existence checking operation is done by inserting the element to each of k hash functions to get k array positions. Then, check whether these k bits are all set to 1 or not. if the bit corresponding to the generated position is found to be zero then the element is not in the set and it is rejected. But if they are all set, then the element is probably in the set.

The spell checker uses the bloom filter algorithm to scan the text the user entered and extracts the words from it. Then, for each word the user enters, the bloom filter algorithm will perform the lookup operation in the loaded dictionary contents to check for the number of occurrences of that word in the array and to check whether the word is there in the dictionary or not. If the word entered by the user is found, the spell checker will do nothing. But if the word entered by the user isn't found, the spell checker will show the mistaken word, choose the closest right word based on the number of occurrences in the array and

it's edit distance and suggest it to the user with the probable correction. The word with few edit distance and more occurrence will be the most probable suggestion.

4. Functionality

The program mainly performs dictionary generation and basic spelling check by using bloom filter MD5 hashing methodology.

To Configure the C:\ "eclipseworkstation " \SpellingChecker1.0.4\resources\config.properties

Change path for proper running and dictionary database creation. Jar file is available in the folder

Path = c:/Users/mikya/eclipse-workspace/SpellingChecker1.03/datafiles/

5. Application Flow Chart

Generate Dictionary >> **Browse File Input** >> **Check Input VS dictionary** >> **Suggest** >>**Generate correction**

6. Conclusion

In this project we use bloom filter with MD5 hash function that operates on correcting spelling mistakes from the given inputs. In future works we can improve this by making the spelling correction in line while putting input with suggestions and grammatic corrections. Even though the MD5 hash has known security flaws.

7. Reference

Inspired by the following reference source
https://www.researchgate.net/publication/305652641_Performance_Analysis_of_Bloom_Filter_with_Various_Hash_Functions_on_Spell_Checker
https://github.com/Connor-Grehlinger/bloom-filter-spell-check
https://github.com/cmesher/Spell-Checker/tree/master/using%20bloomfilter
https://www.cs.dal.ca/sites/default/files/technical_reports/CS-2002-10.pdf