COSC343: Assignment 1 report

Daniel Prvanov (6846004)
August 10, 2022

1 Introduction

Assignment 1 was making a Wordle solver. To help with this each guess that was made would provide results about the guess. it would return the guess_counter, letter_indexes, and letter_states. guess_number is the guess number, letter_indexes is the index of each letter corresponding to their position in self.letters, and letter_states returns either -1, 0, or 1. -1 means the letter is found in the solution but not this spot, 0 means the letter is not found in the solution, and 1 means the letter is found in this spot in the solution. Using these results I was able to use the process of elimination to shrink the amount of potential words the solution could be.

In the rest of the document I will go into further detail as to what I did to help build the Wordle solver, going over my approach, results and an analysis.

2 Approach

The main approach that I used was the process of elimination, eliminating words that could not be correct. The first thing the program does is loop though each individual letter of the last guess, checking if its state is either -1, 0, or 1. If the letters state is -1 the program would use re.findall() to find all words that contained the character. It then checks if the character is in the word more then once. If it is, it would only return these words. If the character was in the word only once it would remove all words that contained the character in the specific index that it appeared in the guess from the re.findall() results. I had to do it this way as I could not figure out how to only delete words that had the character in the original index, instead it would delete all words that had the character in either index. If the letter state is 0 the program will remove all words that contained that letter only once using re.findall(). And if the letter state was 1 the program will use re.findall() to find all words that contained the specific letter. If the word contained the specific letter more then once it would return this list, otherwise if it only appeared once it would return only words that had that letter in the specific spot. After the first guess the program finds the most common letter available in the list of possible words and selects its next guess to be a word that contains this letter.

I imported three libraries to help me create the Wordle solver. They were re, numpy and Counter from collections. Re allowed me to use the re.findall() call which I used to find words that contained specific letters. Numpy allowed me to randomly select a word from a list of words, which I used to guess a word. And Counter form collections allowed me to count the most frequent letter in the list of words, which I used to help choose a better next guess.

3 Results and Analysis

When running my final code with 100 games, easy mode, English, 5 letter words, and 6 guesses the average score my program got was 6.47

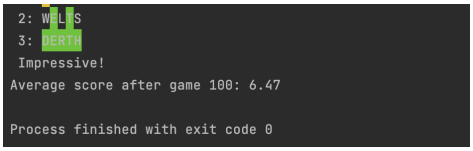

Figure 1: Wordle Score.

I'm almost certain that the reason I was not able to get a consistent score below 6 in English is because I wasn't able to figure out how to deal with when a letter had a state of 1 or -1 but also was in the word more then once. In my implementation if a letter showed up more then once the solver would only keep words that contained that letter instead of also limiting the words to if it was present or not present in a specific index position.

I did not notice much improvement after implementing the letter frequency counter, which was used to select a word that contained the most popular letter in remaining possible_words list. Even though I did not notice an improvement, in theory there should be an improvement as by having the most popular letter be in your guess word, it is the most likely letter to be correct and if it is not you eliminate a large portion of the words in the list.

Language and Mode	Score
English Easy	6.47
English Hard	6.58
Francais Easy	5.55
Francais Hard	4.95
Deutsch Easy	5.59
Deutsch Hard	5.14
Italiano Easy	5.35
Italiano Hard	5.25
Espanol Easy	5.72
Espanol Hard	5.33

Table 1: A comparison of how each language preformed .

From the above table we can see that English preformed the worst out of all the languages by a decent margin. While Italino scored the lowest for easy mode and Francais scored the lowest for hard mode I'm not sure why English preformed the worst but if I was to guess it might be because English might have more words that contain the same letter twice. Also to note about the above table hard mode tended to lower the score instead of increasing it, as we can see that the score decreased in all languages apart from English.

4 Conclusion

So in conclusion I used the process of elimination to create my wordle solver. Overall I definitely enjoyed this assignment, as we had to complete a task that has practical use and I gained a lot of experience and understanding about the Python language and different algorithms.