CS3055 - Mini Project

Solving Wordle in 5 or lesser guesses

Guide: Dr. N. R. Aravind
Name: R. Gokul Kannan
Roll No.: CS19BTECH11048

Introduction

Wordle is a word game developed by Josh Wardle and owned by The New York Times Company in 2022. The objective of the game is to guess a 5 letter word, within 6 tries. After each feedback, the player gets to know if the letters they guessed are in the right place/belong to the word/does not belong to the word. There were 2 different approaches, that were pretty easy to understand, to solve wordle in 6 attempts. But we wanted to try and solve it in 5 attempts.

Idea 1:

My first idea was to do the exact same thing we did for 6 attempts, but reduce the number of letters in the alphabet, so we can solve it in 5 tries. This method could work for approx. 87% of the input words. I just found out the letters with the least occurrence in the 5 letter word dictionary and removed them from the alphabet.

Idea 2:

This idea was to implement a decision tree kind of thing, where we input a word, based on its yellow (correct letter, incorrect position) and green (correct letter, correct position), we decide what word to next input. Entropy is used to determine which word makes the most efficient guess. For now, the decision tree is built for 2 levels.

First level: AESIR

After this if A matches, add 16, if E matches, add 8, if I matches, add 4, if R matches, add 2, and if S matches, add 1. The total is the path you should take for the second level. Based on this id, input words from the following.

Second level:

0	CULTY	**16**	OCTYL
1	MULCT	**17**	CLOTH
2	CUNDY	**18**	OCTYL
3	BUNDT	**19**	PUNTO
4	CYTON	**20**	PLONG
5	PLONK	**21**	PLONK
6	COUNT	**22**	MULCT
7	THONG	**23**	BLUNT
8	LOUND	**24**	BLOND
9	POULT	**25**	BLUNT
10	DONUT	**26**	MULCT
11	POUTY	**27**	PLUCK
12	BLOND	**28**	CLINT
13	PLANT	**29**	BLAND
14	BUNDT	**30**	DUMPY
15	COMPT	**31**	-

Procedure

Python was the programming language that I used. Visual Studio Code was my preferred editor. First we got the file containing all the valid 5 letter words in

English. Next to calculate the entropy of a word, its distinct letters are chosen, arranged in alphabetical order, and 1 to the right most, 2 to next,...and so on till the last is assigned. We iterate through all the words of the dictionary, and find its intersection with the current word, and add it to the corresponding subset. If 4 distinct letters - 16 subsets, if 5 distinct letters - 32 subsets. Then, the entropy is calculated by sum of - (subset/total) * log (subset/total). By this method, I found out the entropy of the word AESIR came out to be the maximum, approx. 3.187.

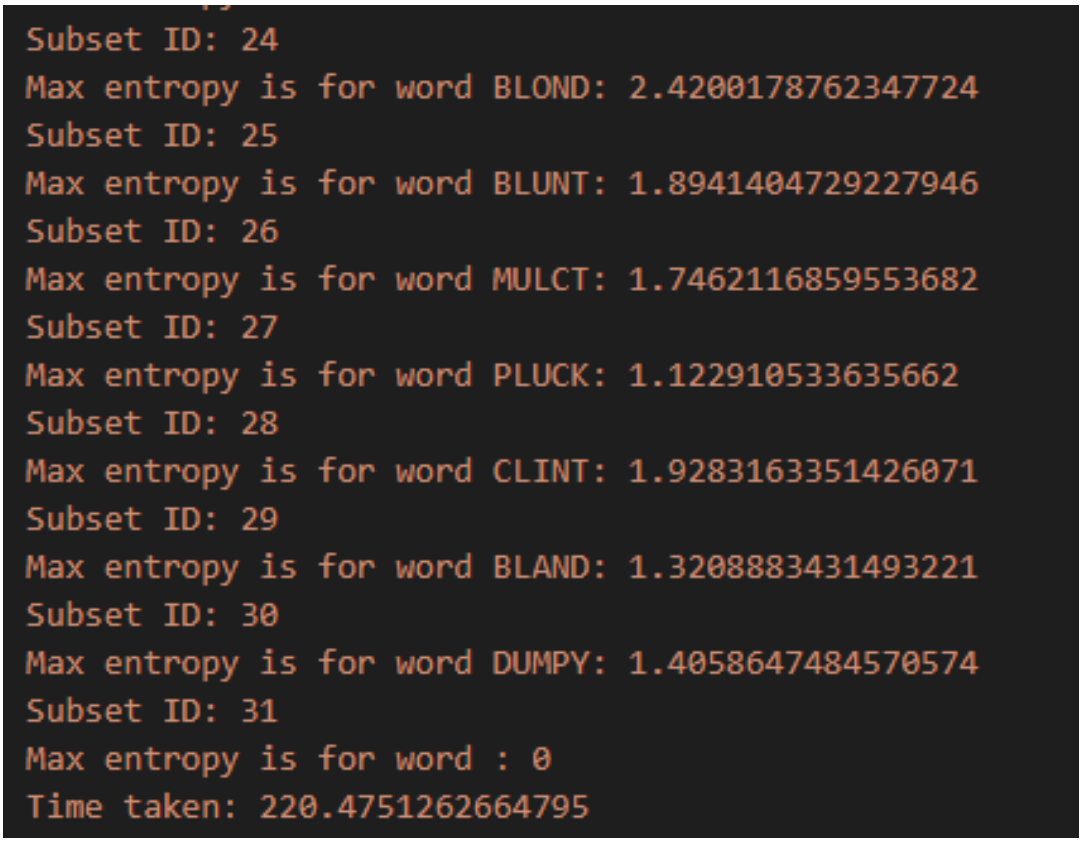
Output

```
Time taken: 229.9267909526825
Max entropy is for word AESIR: 3.1866554283772484
```

Output of the code to find the first level word - i.e. the word with the maximum entropy

```
Wrote 32 files!
Time taken: 2.6129980087280273
List of intersections:
0: 597
1: 916
2: 295
3: 303
4: 553
```

Output of the code to create 32 files - one for each subset of intersection with AESIR



Output of the code to find the second level words - the 32 words

Improvements

The way of calculating entropy for a word can be improved. Here we are doing the same thing again and again (calculating number of intersections for “AB” for example), instead, they can be done in the start, stored in a file, accessed whenever needed. I tried generating files with count of all 1 letter intersections - A to Z, with count of all 2 letter intersections - AB, AC, ..., YZ, with count of all 3 letter intersections - ABC, ACD, ... , XYZ. The next step was to calculate the 4 letter intersections in an efficient way as sir said, but I could not quite get the method.

Further Idea:

Instead of having 32 branches for a node, we can have 11 branches. That is, merge that 32 subset of words into 11 subsets, in a way that each of them have almost equal number of words. If we do like that, in 2nd level 11 words, in 3rd, 11^2 = 121 words, in 4th, 11^3 = 1331 words, in 5th, 11^4 = 14641 words. But in the 5 letter words dictionary, there are only 12478 words, therefore, theoretically, within the 5th row, all words should be present. Hence, we can solve wordle in 5 attempts, if all the words are present in the 5th row. We can try that.

Conclusion:

I came to know about wordle because of this project, and I shared about it with my friends, and asked them to play too. It was quite fun, trying to solve it daily, competing with my friends. I also learnt about the actual applications of the entropy function. I learnt some python programming techniques I was not aware of before. All the programs I wrote can be found in this GitHub link - https://github.com/gokulgk-9402/MiniProject-Wordle . I thank you for giving me the opportunity to work on this project. It was really a good experience.