COSC343 Assignment 1 - Wordle AI

Report for Shay Stevens, ID #7196262

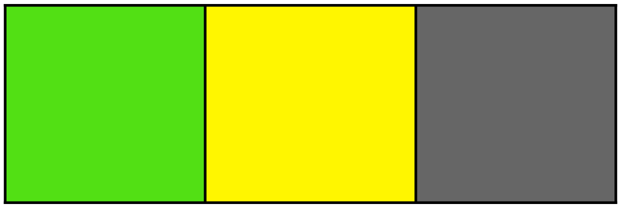
Introduction

AI is extremely fascinating. AI is believed to be the next step of evolution for humanity and it's unsurprising given that the basic AI we have now can effortlessly solve the most complicated problems. To see AI's algorithmic superiority over humans you can look no further than games. Chess is a game widely considered to house the most brilliant intellectuals in the world, honing their mastery of chess over their lifetime. But *AlphaZero*[1], a chess AI, was able to master the game within four hours. This dominance doesn't only appear in chess but within a diverse range of games. Wordle has been a popular game for quite some time now. The basic premise of the game is that a word is chosen from a dictionary and a player tries to guess the word in as few attempts as possible. This report is about an agent I have created in python that attempts to win Wordle with the lowest score possible.

Agent Explanation

My agent is very basic to understand while also being extremely effective. There are two key parts to understanding my agent. Firstly it's logic for filtering to find possible words. And secondly how it chooses a word to use as a guess.

Finding for possible words



There are three states for a letter in a guess.

(Green) State 1 - The letter is in the word and is in the correct position.

[1] https://www.chess.com/terms/alphazero-chess-engine

(Yellow) State -1 - The letter is in the word however, it isn't in the correct position
(Gray) State 0 - The letter is not in the solution

My agent goes through every word in the dictionary and adds invalid words to a list (words_to_remove) in three stages based on the three states.

State 1 logic:
A word is invalid if it doesn't contain a state 1 letter at the correct index.

State -1 logic:
A word is invalid if it contains a smaller quantity of the partial letter or if the partial letter is at the same index as the previous guess.

State 0 logic:
A word is invalid if the state 0 letter is in the word and it is not a partial or correct letter. Or a word is invalid if it contains a state 0 letter at the same index as the previous guess.

My agent then goes through every word in the invalid word list (words_to_remove) and removes every instance of the word within the dictionary then returns the possible words (dictionary).

Choosing a word to guess

My agent has two options for picking a word to guess:

1. A frequency based guess
2. A random guess

My agent will make a frequency based guess unless it knows every letter apart from one then it will be a random guess.

Choosing a word based on frequency

My agent goes through every letter of every word in the dictionary counting the amount of times each letter appears. My agent then chooses the top letters with respect to the word length. Then it goes through every word in the dictionary and gives a score to each

[1] https://www.chess.com/terms/alphazero-chess-engine

word with respect to how many of the top frequency letters it contains. The word with the highest score is then chosen to be used as the guess.

Choosing a word at random

My agent will just choose a random word from any index in the list of possible words (roundDictionary)

Easy and Hard mode

I did originally have plans to use different strategies for easy and hard mode however, my easy mode ended up being far worse on average so I scrapped it. You can still see remnants in the my_agent file of unused functions. My easy mode strategy was to only remove words if they contained letters that I knew you wouldn't be in the solution (state 0) then only guess when you know every letter in the word. It ended up failing because of the limited number of guesses you can make and due to the penalties that incur if you are unable to guess the word.

Analysis:

Best score with default settings

The best score my agent has had wirth the default settings was 4.69. Due to the slight randomness of my agent the average score varies from 4.69-5.7. I have commented out code (lines 546-551) that had a constant average score of 5.06 but I decided to change my agent to have some slight randomness as it is better on average.

Different languages

My agent works well and at times better with different languages than English. This is very impressive considering I am only able to understand english but my agent is able to play Worlde in multiple languages. My agent is also faster in different languages due to the libraries being smaller not making the time complexity as bad. You can clearly see that when the dictionary becomes smaller the average score will become better

[1] https://www.chess.com/terms/alphazero-chess-engine

This was done in hard mode, word length 5, number of guesses 6, seed 0 for 100 games.

Language	English	francais	deutsch	italiano	espanol
Average score (Out of 3 attempts)	4.92	4.35	3.88	4.03	3.71
Dictionary length	11423	5031	3196	3724	2291

Different word lengths

My agent is robust so it can play Wordle no matter what word length however, as the word length gets longer the time complexity starts to become unbearable this is especially obvious with english and word length 10. This will be due to the multiple times my agent has to loop through every word and every letter in the dictionary. For word length 20 in English my agent's time complexity is acceptable and average score is 2. But when word length is 1 or two

Different number of guesses

My agent gets a better average score the more guesses it is given. This is because it will eventually guess the correct word by removing the words that are invalid. The penalty really punishes my agent.

Different seeds

My agent can play well even when the words are random, however, I have noticed that it can be very bad when the words are random. The average score is usually better 5 - 6. It is better to use (lines 546-551) when the seed is random

Conclusion:

My agent is a pretty solid wordle AI with some obvious problems that can be improved

[1] https://www.chess.com/terms/alphazero-chess-engine

upon.

Things I could improve

* Optimizing the time complexity. It is painfully obvious that my code is too time complex due to it taking roughly 3 minutes to run on my computer this could easily be improved by writing simpler code with less for-loops

* Different strategies with easy and hard mode. I made an attempt to make different strategies for both however, my easy mode had a worse average score as alluded to earlier. Easy mode should definitely be better on average due to the ability to still guess words that don't have letters present in the solution. A better strategy than the one I attempted to implement should be done.

* Using entropy based guesses instead of frequency and random. I have done research into creating a good Wordle AI and have found that an entropy based guess is better on average than a frequency based guess. This could be used to create an agent with a better guessing strategy than the one I have built.

Notes

My code was run using a computer with a i7-10700KF and it still took around 3 minutes to complete English with 5 letters. I was unable to run it on the lab computers but since it is complex on my computer I know it will be slow on the lab computers hopefully not too slow.

[1] https://www.chess.com/terms/alphazero-chess-engine