WordleBot

Research Computing Coursework

Ben Boyd

19th January 2023

Abstract

This project investigates the use of information theory in order to solve the game Wordle using a WordleBot. The bot maximises the entropy of the next guess in order to give the highest expected information. The starting word "salet" is found to average the fewest number of guesses when compared to other common starters such as "crane" or "audio". This is consistent with literature that finds "salet" is statistically the best starting word, despite not being a possible final Wordle answer itself. During the devlopment of the WordleBot, the cut-off point at which it should stop guessing from the larger set of guess words and only choose words from the remaining small set of possible words is considered. It is found that the optimum point when using the word "salet" is when there are five possible words remaining. This yields a minimum average number of guesses of 3.448, which is only 0.027 off the best possible bot performance. It is found having an earlier cut-off point leads to a larger standard deviaton of total guesses, regardless of starting word. Profile testing finds these games can be completed on average in 3.8 seconds at 0.34 seconds per guess when starting with "salet", due to the precomputation of the every best second guesses. It is concluded that it would be possible to improve accuracy marginally by planning two steps ahead, however, this would come at a cost to execution time.

1 Background and Motivation

In October 2021 Josh Wardle released the game *Wordle*. The game was simple but soon captured the attention of hundreds of thousands of players who would attempt to find the word of the day. The popularity of the game led to it being purchased by New York Time's by an undisclosed seven figure fee in January 2022. At the peak of its powers, Wordle's dedicated players would regularly debate what the best starting word is order to achieve the lowest average number of guesses. This problem naturally caught the attention of statistician's and data scientists who attempted to find the formula for the perfect game.

The laws that govern Wordle are relativelyt simple. There is a five letter word the player needs to find in the fewest number of guesses. After each five letter the guess, each letter tile will turn a colour depending on whether it is presents in the true word. Grey tile means the letter is not present in the true word at all, yellow tile means the letter is present but not in this position and green tile means the letter is in the correct position. Double and triple letter rules are sligtly more complex, where only one of the double guess letters will have a colour if it appears once in the true word. The same rule applies if triple guess letters appear once or twice in the true word. The first occurrences of the correct letter are normally given the coloured tile, unless there is a green tile later on in the letter. After you know these rules you are able to simulate the outcomes of the Wordle tiles, just like the website, for any pair of words.

Being able to simulate a game of Wordle is useful, however, since there are over 158,000 five letter words in the English language, it would still be difficult to computationally find the formula for the perfect game. Furthermore, a thorough statistical analysis would require an understanding of how common words are in the English langage. How much more likely is the word “hello” to be used than the word “ouija”? Luckily this does not actually need to be considered as the official Wordle word lists were conveniently found in the source code. The first list uncovered is a list of 12,947 five letter words that Wordle will accept as a guess. The other list is a subset of 2,309 of these words that are used as possible true Words. Interestingly Wordle moves one place down this list each day meaning each possible answer word has the same probability of being today’s word. You could even cross words out of this list if you know words that occurred the days before, however, out of fairness most ignore this.

Given that the word lists are known and it is possible to deterministically predict the outcome of any Wordle game, what is the best way to play the perfect game? The secret to this is picking the guess which provides the most information on what the true word might be. This is a concept of information theory.

In the context of Wordle, the new information is not given by the guess word, it is given by the tile pattern that it produces. Information theory defines a single bit of information I as a tile pattern x_i that can reduce the number of possible remaining Wordle answers by a half, given guess word g. The conditional probability of getting a pattern given a guess as word is defined as

$$p(x_i|g) = \left(\frac{1}{2}\right)^I \tag{1}$$

This equation can be rearranged to make information the subject

$$I = -\log_2\left(p(x_i|g)\right) \tag{2}$$

This tells us that the pattern combinations that are least likely provide the most information. It is true, however, that when selecting a potential guess word g it is not possible to know for sure what tile pattern it will produce since the true Wordle word of the day is unknown. For this reason the expected information $E[I]$ must be considered across all N unique tile patterns a guess can yield

$$H = E[I] = -\sum_{i}^{N} p(x_i|g)\log_2\left(p(x_i|g)\right) \tag{3}$$

This expected information is equal to entropy. It is this quantity that needs to be maximised in order to pick the best next guess.

It is possible to take Eq.(3) further and included the expected information of the best next guess in the calculation of the expected information for the current guess. In other words, it is possible to plan multiple steps ahead when determining your next guess. To consider every future step when choosing the first guess is incredibly computationally expensive as it requires simulating every possible game of Wordle. This does not only mean simulating every Wordle answer,but also simulating every guess word at every step before a single move has been made. Despite the computational complexity of this problem, it has been implemented. Bertsimas and Paskov (2022) showed that the optimal starting word is “salet”. They optimal classification trees to showcase their policy which yields an an average of 3.421 guesses to solve Wordle. This is to date the best performance on Wordle achieved using a set of algorithmic rules and it is unlikely to be beaten.

Some might argue that it is best to take a machine learning approach that would not require the same level of statistical computation or rigor. Bhambri et al (2022) used reinforcement learning algorithms that learnt the optimal policy by repeatedly playing Wordle games and learning from it. Although this did not require the same high levels of computation as analytical solutions, the approximations meant on the reinforcement learning solution had a higher average of 3.508 guesses.

This project will take a light-weight analytical approach where the WordleBot will maximise information based on the next guess without planning ahead. The objective of the project is to minimise the average number of guesses needed to solve the game. Special attention will be focused on the optimal point to stop guessing from the guess list and guess words from the remaining possible answers. The project will also aim to make guesses quickly in real time, so that the Bot can be used to solve a live game of Wordle. The report will firs outline on the methodology used, then it will look at the development of prototypes and will finally comment on the accuracy and efficiency of the best implementation.

2 Method

The model makes decisions only by calculating the expected information gain for the next guess word using Eq. (3). In practice this is done considering every possible Wordle pattern where each of the 12,947 allowed guess words are used to guess the 2,309 possible Wordle answers. This relies on simulating the tile outcomes for nearly 30 million different guess and true word pairs. The results of these tile outcome patterns are stored in a compressed three dimensional array that can accessed quickly by the WordleBot. To calculate the expected information gain by picking each guess word, the probability of each unique pattern given a guess word is calculated. This involves counting the number of possible true words that would give a certain unique pattern given a guess word and dividing it by the total number of possible words remaining. The probabilities of each pattern are converted into the expected information of a guess using Eq. (3). This process can be repeated for each guess quickly by considering a two-dimensional guess slice of the three dimensional recomputed tile patterns array.

Since it has been statistically proven that the first guess word that provides the highest expected information gain is "salet", the WordleBot always start wit this guess word by default unless instructed otherwise. The WordleBot will then wait for the tile pattern response for this starter word, given interactively by the use or simulated if the true word is known. Once the pattern is given the bot will remove potential answer words that are not capable of producing this tile pattern given the first guess word. Since these impossible words are also removed from the tile patterns array, the next computation of the expected information gain of each guess word is different. The WordleBot will then pick the guess word that maximises the expected information gain. This process is repeated until there are only a few possible words remaining. Since "salet" is the default first guess word it is useful to pre-compute the best second guess word corresponding to each possible pattern the first guess can produce. This increases time-efficiency of the bot as it is only calculates the expected information of each guess once the possible word list is substantially reduced.

After the possible word list has been substantially reduced there comes a point where the algorithm should only make a guess from this small list and not from the list of 12,947 allowed guesses. It obvious to swap to this list when there are at least two words remaining since one guess will eliminate the other. It could also be argued that if there are three possible words remaining the next guess could be chosen carefully based on expected information so that if it is wrong it could eliminate another possible answer as well as itself. This logic can be applied to when there are four or five possible words remaining, but it not obvious where the best cut-off point is to guess only from the smaller list. This cut-off point is investigate during the development of WordleBot.

3 Development

The first stage of development involved implementing an algorithm that would accurately predict what tiles a guess word would produce for a given true word. This was non-trivial as it relied on getting the tiles correct for guess words that have duplicates or three of triplicates of a letter that only appears once or twice in the true word. This meant that the algorithm had to cycle through each guess letter at least twice as it is not possible to know whether a tile should be yellow or grey before reading the full word. After dealing with these tricky cases, unit test were constructed so it could test whether the right tiles were being produced for all the sensitive words

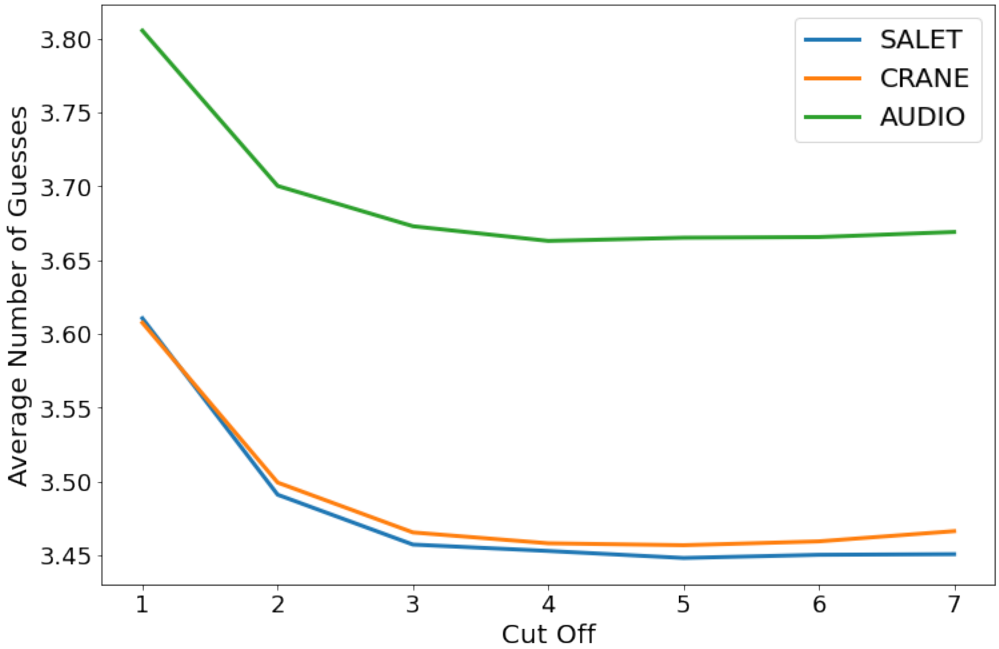

Figure 1

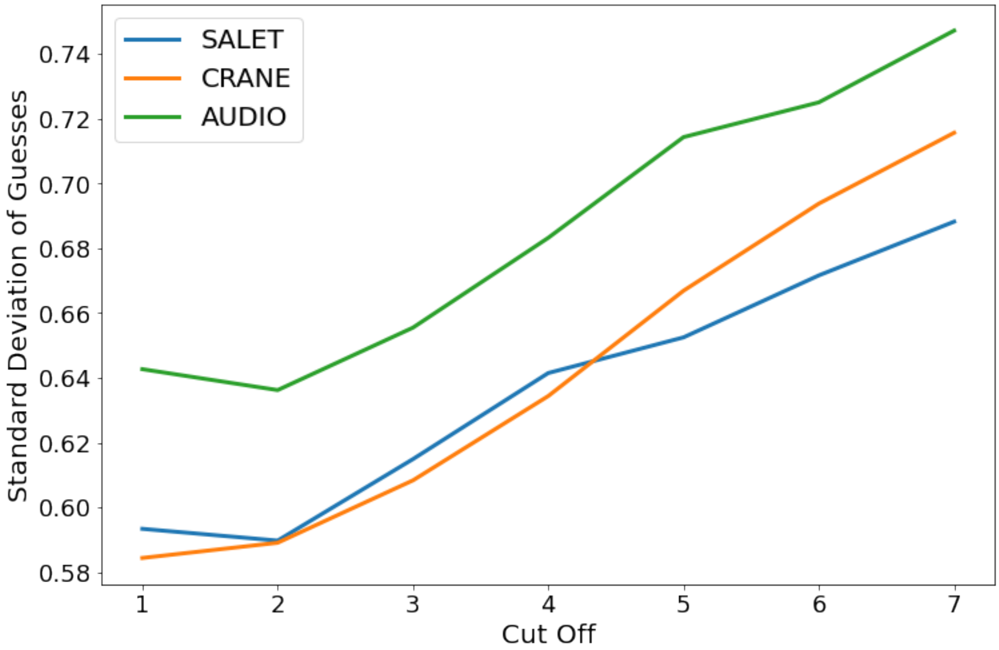

Figure 2

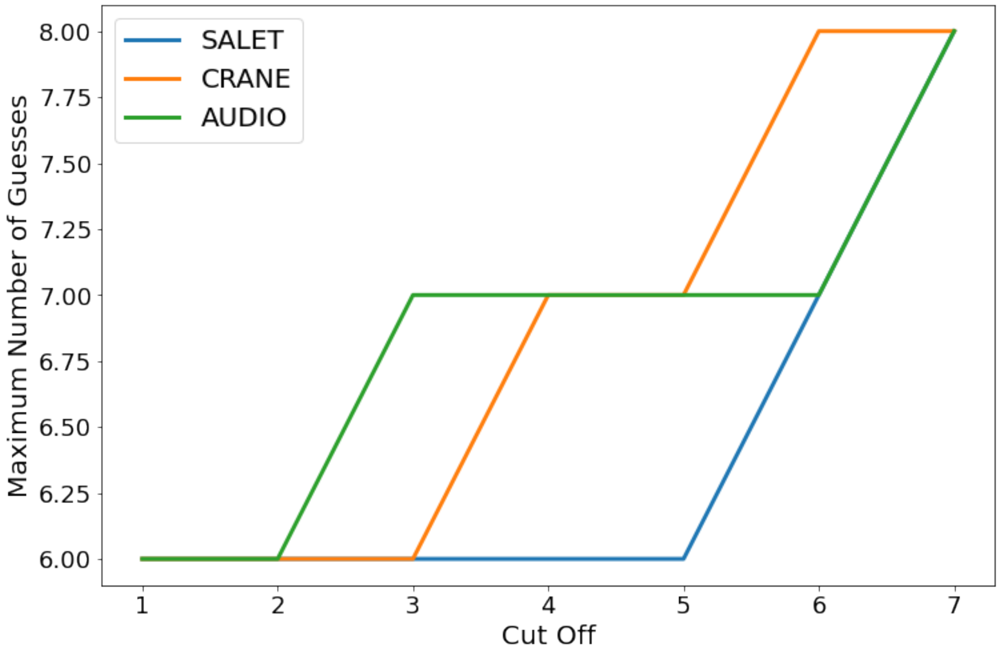

Figure 3

4 Final Results

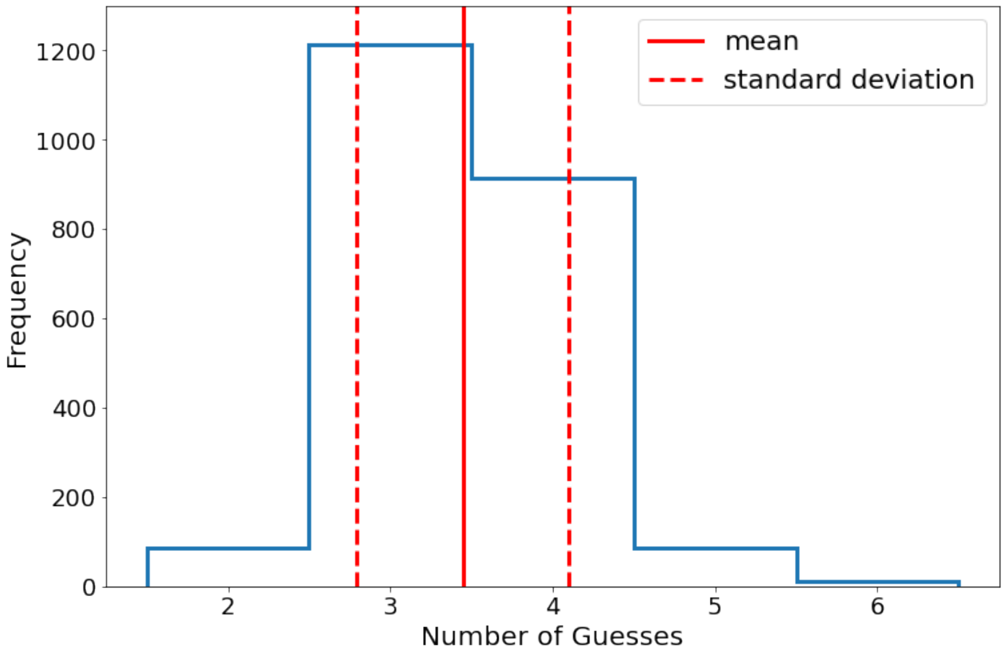

Figure 4

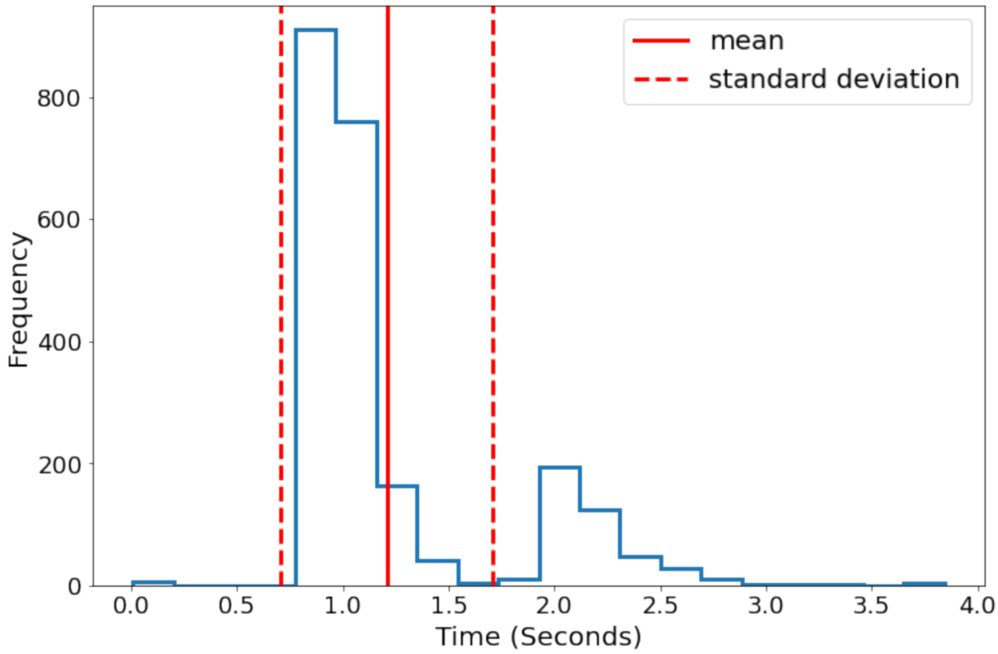

Figure 5

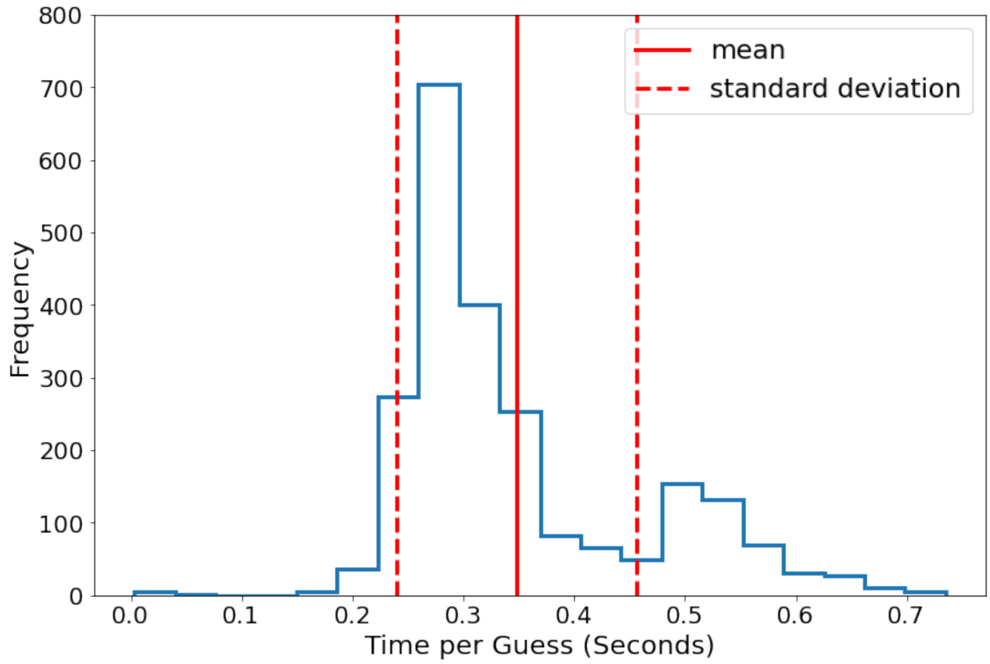

Figure 6

5 Conclusions