Solving a Wordle

Terry Cao (cao00111@umn.edu), Maxim Zabilo (zabil003@umn.edu)

April 2022

Abstract

Wordle is a web based word game that has spiraled into popularity this past year in January 2022. The players must determine a hidden word with 6 guesses. With each guess a player is given new information to help aid in their search for the final answer. There is lots of discussion online about what the best starting word is for Wordle and their have been countless algorithms made to beat the game. We wanted to develop an algorithm that could solve the Wordle of the day and determine what the best starting word would be. We found the best word to start with that will use the least amount of guesses as well as the best word to solve the most Wordles using our algorithm.

1 Introduction

1.1 What is Wordle

Wordle is a web based word game that prompts users to guess a five-letter word with a few constraints. The constraints are that it must be a five-letter word, only utilize the 26 letters in the English alphabet, users must guess the word in 6 attempts or less, and utilize given hints based on a color coded system [10]. The color coded system determines whether a guessed word has letters in the Wordle and if they are in the correct position. A letter is marked green if it is in the Wordle and is in the correct position. The letter is marked yellow if it is in the Wordle but is in the incorrect position. The letter is marked grey if the letter does not appear in the Wordle at all. The player must guess the Wordle of the day in six guesses. If the player fails to guess the Wordle in six guesses then they have lost the Wordle of the day. Players have the option to play on hard mode which adds another constraint that enforces the player to utilize green and yellow letters in their guesses. Currently, the Wordle solutions are a predetermined set list that has been hard encoded. This means previous Wordle solutions from past days will not be solutions in future games.

Wordle has become a global phenomenon that has captured the interests of the younger and older generation. New York Times even went out of their way to purchase the Wordle game from the developer and began publishing it themselves due to the incredible surge

in popularity. Both of us began to participate in the daily Wordle as soon as it rose in popularity. Naturally as programmers, we thought it would be interesting to see if an algorithm could solve any given Wordle within the given constraints. We found Wordle to be slightly competitive in the sense that if one is able to determine the Wordle in the least amount of guesses they were "smarter". What if we could encode an algorithm that not only solved the daily Wordle but it also did so in the least amount of guesses. Also there is a big debate in social media as to what is the best starting word. Is it one that contains many vowels or one that prunes the most letters?

1.2 Problem Description

We want to develop an algorithm that can solve any Wordle. First, we need to evaluate if it would be a better choice to have the first guessed word chosen randomly or not. We determined it would be best to choose a starting word not randomly but based on a score it will be given by an algorithm. The algorithm will give a word a score based on the frequencies of letters in the word [3]. The algorithm would have to start by combing through all the guessable words and determine the letter frequencies in each position. There is a full list of five letter words that comes in at a total of 12,966 words. There is also a pool of guessable words that comes in at 10,657 total words and a sub-list of winning words comes in at 2,315 total words[2]. The winning and guessable words are the possible solutions and guesses to a Wordle game as officially documented by the creator himself. Our approach will differ if we end up using the full-list or the guessable pool of words. The controversy arises if we want to make the algorithm more human-like or not with complete knowledge of all five letter words in the English dictionary. We opted to use the reduced guessable list to mimic the creators intended way of playing the game, which had the intent of pruning words that most people didn't know.

Next, we need to decide how we want to store the words in our algorithm. A sorted word list saves a lot of time, but as for whether the word string should be stored alphabetically or based on how the letters in the words sit relative to each other is debatable. Another way to store a word is by keeping an array of length 26 which takes one spot for each letter in the alphabet. If a spot in the array is a 0 that means the word doesn't contain that letter, otherwise put a counter for how many of that letter the word has in the correct spot in the array (where indexes 0-25, correspond to a-z, ex: 'crane' is stored as 1 0 1 0 ...). Another approach could be to create a Word class. This class would have a list of the letters in a word and an int score. The score determines how likely it is to be the Wordle. For now this score is evaluated by adding points based on number of correct letters, and correct letter positioning. This way we'll sort the words by score and also prune words with scores of 0 meaning they contain letters not in the Wordle.

Lastly, the algorithm itself will need to be set up in such a way that prunes words out of the pool of possible guesses most efficiently. With the first storage approach a word will be

taken out if it contains letters already known not to be in the word or are not in the correct spot. For the Word class approach the word would be pruned if in the current state of the game it is given a score of 0. If our algorithm is good enough we should be able to replicate results of [4] with the word being guessed 100% of the time with 5 or less guesses.

2 Background

2.1 Word List

For the game of Wordle there is a solution word list consisting of about 2,000 words, but there is also a guessable word list being much higher at around 13,000. The list of Wordle solutions will be based on this source [11]. Other articles either make their own lists of 5 letter words based on the English alphabet or use the official one. We will be using the official word lists provided. This article by Arvi Hurskainen goes over how the length of the Wordle which is 5 for English has to be vastly different for other languages because of the morphological differences [4]. While not pertinent to the problem at hand it does bring up a potential question to explore, does changing the acceptable words list numbering at 12,972 to a different number make the challenge of solving a Wordle easier? Nisansa de Silva performed studies on character statistics on 466k English words [3]. This lead her to determine one of the best starting words was "raise" which overlapped in another study she cites in her paper. This leads us to believe that list size does not matter to reinforce our usage of the smaller guessable list.

2.2 Initial Word Choice

Throughout our search for articles we found that a vast amount of them focus on the first word that is chosen for a game of Wordle. This doesn't seem surprising since the number of guesses it takes will vary greatly depending on how much information a word can provide us. What word to start with? First cut the length of words to length 5. Then create a letter frequency map. Make a unique word list with words that have 5 different letters like "sport" but not "feels". Scores will be assigned to each word based on the character frequency mapping and then greedy best first search is used to find the best starting word. Based on overlap of characters from the word with the highest score other words are removed until 3 words are found with the most coverage for first guesses. An interesting observation is that the letter 'a' is more frequent than 'e' in the list of Wordle words which isn't the case for the whole entire English language [3]. Nao Lubin presents the word 'arose' as the best starting word based on the letter frequency and heat maps generated from the 5 letter words from the English dictionary. To fully calculate the probabilities of a words appearances, a tree can be constructed based on the position of the letter and whether that letter will be assigned green, yellow, or grey color for that position. Then the word with the lowest entropy is chosen [7].

There were several different initial starting words presented by the various authors of the papers we reviewed. Johnathon Olson claims that it's possible to solve 99% of all puzzles in only 4 guesses or with an average of 3.42 guesses per win [9]. A good note by Olson is that most of the best starting words discussed online are based upon the effects, eliminations, and entropy of just the first guess [9]. This gives a different take on how to look at determining the first starting word. Here Olsen leans more towards an approach that involves partitioning words into sets based on what colored letters they would get on a given guess. The smaller the sets the more information we gain. Olson determines the best word to start with is "rance" for most solvable words in 4 guesses or fewer. He also found that the best word to use for the fewest average guesses was "salet", which performs at a 3.42 average guess rate. [9].

Throughout all the papers we examined there is a recurring theme that the first guess should involve as many vowels as possible. Since these words with a great amount of vowels make up a big portion of the word list, they are likely to prune the most words in search as well as accumulate the most green letters in the search. Tom Neill determines that "raise" is the best starting word based on his own search algorithm. The algorithm starts by determining how many words in the word list have a specific letter at the beginning of the word, somewhere else, and not in the word at all. This will give you a value that will determine which letter is better than another in all five positions of a five letter word. For example, 'e' is a better starting letter than 'z' since there are 72 words that start with 'e' in the word list and only 3 words that start with a 'z' in the word list. Now to determine a word's value you must combine the value of multiple letters stemming from each letter choice in each position which he represents as a tree diagram. This is reminiscent of A* search since high scoring words are determined by looking at the best path down the tree. This algorithm then prunes words based on low scores and eventually would determine that "raise" is the best starting word utilizing Neill's heuristic. [8].

Even when given an optimal initial word choice for a Wordle that doesn't mean that the rest will be a walk in the park. A lot of articles that we found focused on strategies that could be used to beat the entire Wordle. Although not all of the strategies can be used by humans as we just don't possess computer capabilities, we can still mimic some of the algorithms.

One approach that Benton J. Anderson and Jesse G. Meyer used is reinforcement learning which was done in python using the gym package. Three classes that were used included game, agent, and environment. Q learning was used to allow agent to find optimal policy for playing Wordle based on a reward system. Yellow = 2, green = 5, win = 25, loss = -15. Two tests were conducted and involved 1000 and 10000 games played. Agent could make 5 types of guesses, random, word likelihoods from letter probabilities, green letter probability maximization, smart, or exclude. The random guess picks a word from the word list. This method proved to have a total win rate of 64.8% after 10,000 trials using the best parameters.

Most games lasted all 6 rounds. The following were determined to be the best starting words: "bowne", "slaty", "prick", "faugh" and "meved". The best strategy Anderson and Meyer came up with is to get as many yellow letters possible and then make guesses to exclude more letters with words that contain letters that are unused. If the player receives a green letter, then use all info available and limit the word list for the next guess [1]. Tyler Glaiel Algorithm, used a hypothetical score for each word in the list of possible guesses. The score is computed by comparing every possible guess with every possible solution. Green letters will give two points, yellow awards one point and grey is zero points. He found that "soare" had the highest score and finds solutions in 3.69 guesses. Worst case 8 guesses. The word "roate" got to solutions faster and on average made 3.49 guesses with the worst case being 5 guesses. Mahmood Hikmet, Algorithm deduces frequency at which letters appear in the list of possible solutions and ignores other guesses. Words that use more popular letters in list of solutions are better for starting with. From most effective to least effective starting words: SLATE, SAUCE, SLICE, SHALE, SAUTE, SHARE, SOOTY. Andrew Taylor algorithm, Combines two strategies where each word gets a score and lowest signified best guesses. REAIS is the best starting word since 168 solutions will not have these letters [5].

Not taking the most optimal approach to the problem as the author created their own 5 word list, but it is more for the sake of creating an algorithm which could be used in general, not taking into account any specifics of the letter frequencies of the original Wordle list. Most Rapid Decrease Algorithm or MRD is a greedy algorithm which tries to decrease the word list as much as possible before only 1 or 2 guesses are left [10]. Another algorithm introduced in this article is Greatest Expected Probability or GEP which tries to maximize the probability of the solution being guessed from the next list of words. Both algorithms find that the best word to start with is 'tares' [10]. The algorithms here only cover choosing words step by step but not with the whole picture in mind which could make information gathering even better. Another article used a similar greedy approach [10].

In the included AIMA code from the course we were interested in re-purposing some of the code in the game.py and search.py. These programs define class structures that would be relevant to our Wordle game solver. The game.py program specifically defines a game class that is reminiscent of the expected class that we'd like to encode for our own program. Jack Coop describes an actual encoding of an algorithm for a wordle solver [2]. The algorithm Coop encodes assigns scores to words based on letter frequency. Another repository which can help us build upon the aima code is by Jonathon Olson who explores solving the Wordle problem with search trees which is also something that we are going to do [9].

2.3 Other Subjects of Interest

Wordle can be phrased as a NP Hard problem meaning it can be solved in polynomial time[6]. Lokshtanov goes over an extensive proof for the above statement [6]. Another conclusion reached is that the number of guesses will be less than or equal to the number of

letters in the alphabet which isn't particularly exciting. The run times of different algorithms that are researched above can be compared to see which ones find the solutions in the least number of guesses possible and the fastest.

3 Solving a Wordle: An Approach

3.1 A Break Down

Wordle can be thought of as picking fruits from an overflowing basket that loses fruit each time one is picked out of the basket. When a word is chosen as a guess it constrains the rational choices made for other words chosen as guesses down the line. To show what is meant by a rational choice look at the example below. Before that, however, let's establish an easier representation to read. For guesses if the letter is not in the word it will be represented as N for Gray. If the letter is in the word but not in the correct position it will be represented as Y for Yellow. If the letter is the right one and in the correct position it will be represented as G for Green.

The algorithm we've come up with will take great inspiration from Jack Coops implementation of a Wordle solver. It will pick a starting word and then retrieve a score based in a 5 element array that is based upon the N, Y, G system. Each element represents a letter and position of the correct answer and is assigned a score promptly. The algorithm will assign a 0 for a letter that is 'N', assign a 1 for a letter that is 'Y' and Assign a 2 if the letter is a 'G'. Now that we have a score for each letter we'll parse which letters need to stay and which need to be removed from the guess pool. We'll encode a list that contains the 'G' and 'Y' letters and remove any 'N' letters. Next, we'll encode a function that prunes all guessable words with any 'N' letters. We keep track of all 'Y' letters and utilize a function that will provide the remaining guesses with the 'G' letters in position and potential words with 'Y' in any of the remaining positions. Now that we have a pool of Guessable words the algorithm will continue to guess and parse in valid/invalid letters and their positions and prune words until we've reached a solution. The algorithm will make a guess based on a score that is given by occurrences of a 'G' letter in the correct position and a 'Y' letter that is somewhere in the word. The highest scoring word is then guessed.

3.2 A Sweet Example

For example, if the word "apple" is chosen and the target word is "crepe" then the letters from "apple" would be colored as NYNNG. The letter 'a' is not in 'crepe' so the first spot would be N. The letter 'p' is in 'crepe', but it's not in the right spot so it will be colored as Y. The letter 'p' is in 'crepe', however there's only one 'p' meaning that it will be colored as N because there's already one 'p' colored as Y before it. The letter 'l' is not in 'crepe' so it's colored as N. The letter 'e' is in 'crepe' and it's in the correct spot so it will be colored as G.

3.3 Algorithms and Approaches

3.3.1 Design: Determining the Best Starting Words

To solve a Wordle enough correct choices have to be made so that all the letters light up Green. From a probability standpoint, the best way to choose the next word is based on how frequent it is. First, we had to decide what our best starting word/words would be. With the literature we reviewed we compiled a list that is the accumulation of all the best words found by each author. Those words were raise, clout, nymph, rance, arose, bowne, slaty, prick, faugh, meved, soare, roate, slate, sauce, slice, shale, saute, share, sooty, reais, and tares. We believe these words to be among the best starting guesses based on the literature and our own testing. We want to validate the use of these words with our own tests. First, we want to take each starting word and test how it actually plays when used in a real game of Wordle. The starting word will be deemed valid based on if it can solve most Wordles in under 7 guesses. If the starting word takes more than 7 guesses to solve a Wordle then we will note that specific Wordle was unsolvable by the starting word. We will run each starting word on all 2,315 Wordles and determine the average number of guesses our algorithm takes to get to the Wordle of the day. We will also keep track of the number of unsolvable Wordles which are the Wordles that take more than six guesses to solve for each starting word. This is necessary since our heuristic is likely not going to be perfect and able to solve every single Wordle in under seven guesses.

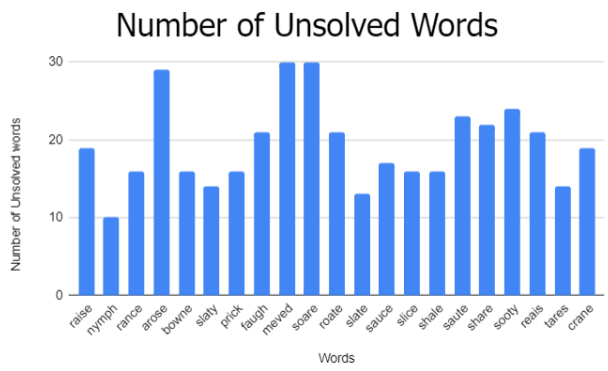
Next, we'd like to run our own tests to determine letter frequencies in the entire guessable word list. To do this we'll run an algorithm that combs through three different word lists. These lists include the guessable word list, the Wordle solutions list, and the full list of five letter words. The algorithm will then have a global counter for each letter in each position of a five letter word. The goal is for the produced graphs to show similarities with letters and positioning in our best starting words.

4 Experimentation and Results

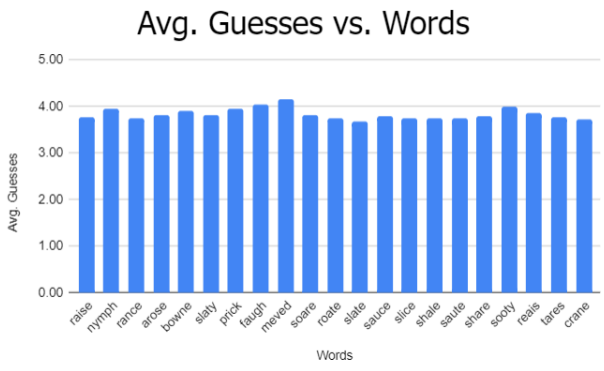
4.1 Best starting word

The algorithm we made was able to solve most Wordles utilizing different starting words. By playing all 2,315 Wordles with the 21 best starter words we have determined that the best starting word with the best average guess rate per solvable Wordle is "slate". The word "slate" had an average guess rate of 3.67 guesses per solvable Wordle as seen in Figure 1, Graph(b). The starter word to solve the most Wordles with the least amount of unsolved words was "nymph" as seen in Figure 1, Graph (a).

Through our letter frequency algorithm, we determined that the most common letters in increasing order of position for the full list was 's', 'a', 'a', 'f', 's' as depicted in Figure 2.a.



(a) Number of unsolved words for each of the 21 best starting words. Note that the unsolved words took more than 7 guesses to solve.



(b) The average guesses for each of the 21 best starting words.

Figure 1: Graphs (a) and (b)

For the guessable list Figure 2.b illustrates that the frequent letters in order of positions are 's', 'a', 'r', 'f', 's'. For the solution list we found 's', 'a', 'a', 'f', 'f'.

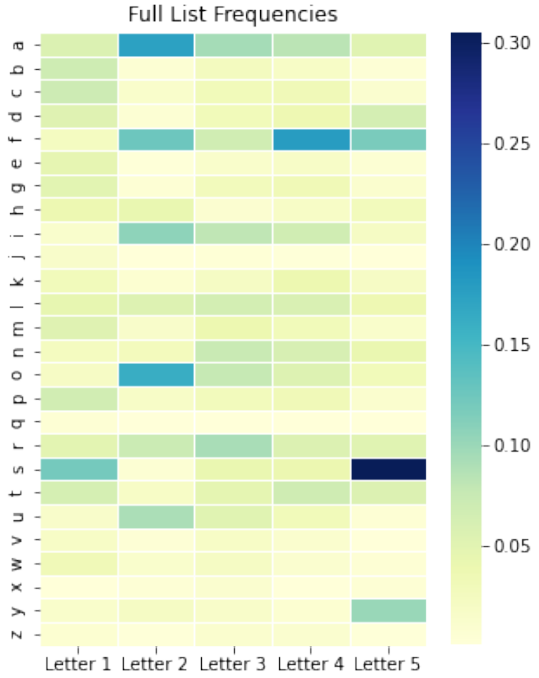
4.2 Analysis

An interesting observation is that words that have multiples of some vowel like 'sooty' are often chosen as starting or secondary guesses. While it makes sense because it's higher up on the frequency score list, it does not make sense from a coverage perspective. A good guess will get as much letters either right or wrong as possible especially for an initial guess. Choosing a word with double vowels is not the best strategy since there aren't too many words where there are multiple of the same vowel. The reason that they're often chosen is because by frequency they have the highest score which is only natural since there are only 5 vowels and 21 consonants. There almost always is a vowel between two consonants except for some exceptions and there has to be a vowel to make a word in English meaning vowels are used in every word and more frequently than the rest of the consonants. Vowels are hence more frequent.

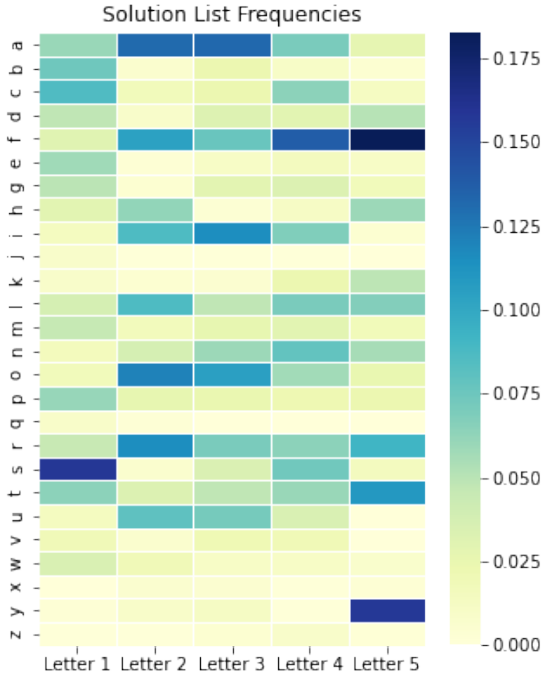
For pairings from Figure3 one corner that clearly stands out is the one with pairings of the letters r,s,t with a and f. In fact it seems that pairings with the letters r,s,t are more frequent than any other letters. For that reason alone the word "crest" could be a good starting guess or "trash".

4.3 Game Plan for Best Human Strategy

For humans the strategies that involve filtering a lot of words just aren't feasible as the average population just does not have the capacity to memorize so many words and recall



(a) Full list = Guessable + Solution lists

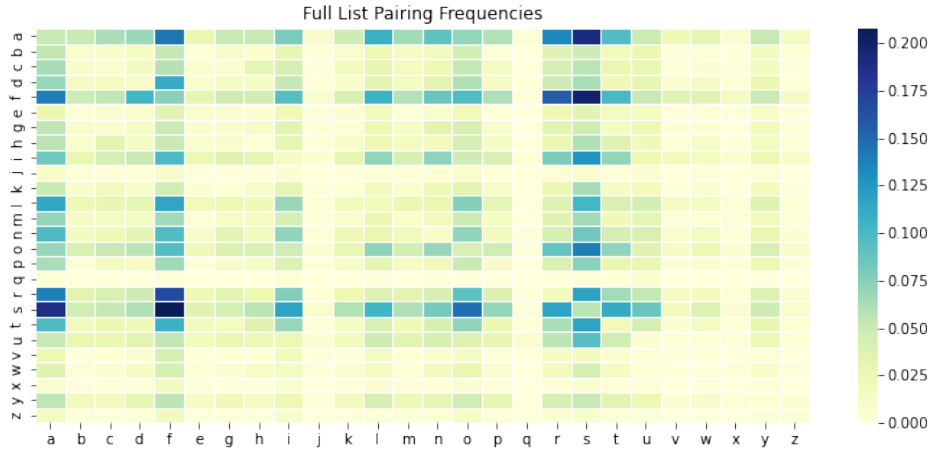


(b) Frequency of S is high in the last spot of the word for the Guessable list

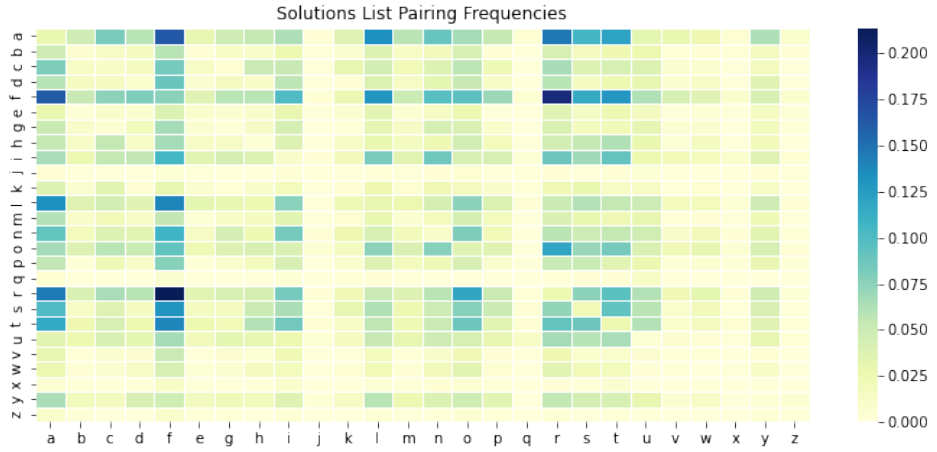
Figure 2: Count of each letter in each spot of a word

them. Instead, of focusing on all the possible guessable words it's nice to have a guessable list that covers all the most commonly used letters found in the solution list. The letters: 'j', 'q', 'x', 'z' are each used in less than 1% of the solution words meaning they should be considered as last options when making guesses. The letters that have the highest frequencies in the solution list are: 's', 'a', 'e', 'o', 'r', 's', 't', 'i', 'l', 'y' meaning they are the letters most likely to appear. It is surprising that 'y' appears in this list, but that is because it is often times the last letter of a word. Rather than trying to solve a Wordle by simulating a computer approach of guessing only words with letters that contain mostly either Yellow or Green squares from previous guesses it is smarter to go for a breadth strategy. Computers are able to search depth-first because of their memory. Since humans do not have such capacity, instead a search to cover the most frequent letters will suffice.

For a breadth strategy the goal is to cover as many letters as possible so that a person will have a limited pool of letters from which to form the word that they are trying to guess. Since there are 26 letters in the alphabet 5 guesses would eliminate 25 letters, if using a variety of words. By that point a person should be able to guess on their sixth try what



(a) Pairings for words in the full list



(b) Pairings for words in the solution list

Figure 3: Pairing frequency of each letter with other letters in the alphabet

the word is or even before that. The words that provide the most breadth in that sense are: "slate", "crony", "guild", "heart", "rusty". Because there is overlap in these words it is not for the best to use them consecutively, but they can be a good start. Out of all the vowels 'u' appears the least so it words like "guild" and "rusty" may not be that great. What matters most is that the vowels in the word are figured out meaning that using "slate" as the first guess followed by "crony" and "guild" give maximum coverage for 14 letters. Which would increase to 15 letters if the word "stare" is used instead of "slate" for the first guess. Give it a try!

5 Conclusion

5.1 Summary

With the algorithms we've developed to determine the best starting words to solve Wordle. We determined that "slate" is the best word to start with for the lowest possible guess rate at 3.67 guesses per solution. The best word to start with for most solved Wordles was "nymph". We determined this by running an accumulated list of best starting words through a letter frequency algorithm. Then the words are ran through all 2,315 Wordle games to produce the statistics previously mentioned. The algorithm to solve the Wordle we developed utilized pruning of words based on a 'G', 'Y', 'N' system that removed words with 'N' marked letters. The algorithm then makes guesses based on the amount of 'G' letters and 'Y' letters.

5.2 Future Work

There are a few different things that can be done for future work. Instead of trying a greedy algorithm a neural network could be used. This is because the weights for different frequency scores can be adjusted for by controlling the amount of points that a word earns when being scored. Right now it's just based on the frequency, if a letter in a word has this frequency the word gets that frequency added to its score. Other weights can be attached to the words based on the grammar of the word. The same can be done for pairing frequencies and the score given to a word based on the game state. What is a word provides good coverage, but it clearly can not be the word because it has one letter that is not in the word trying to be guessed? Should it still be completely discarded?

If a different type of algorithm is used it can have an information gathering stage as well as just a guessing stage once the word list is small enough that it can comfortably guess without taking too many turns.

Since greedy search doesn't know grammar it doesn't know that what's likely to appear after an 's' confirmed as Green. It might be a vowel, it might not be, it's most likely a vowel not a consonant, but the algorithm does not know that.

6 Contributions

Terry and Maxim both cooperated on the code. Terry focused on defining the needed classes as well as the algorithms to determine the best starting words. Maxim focused on encoding the search algorithm and score system. Maxim and Terry both had major and equal contributions to the final report, about fifty fifty in terms of work gone into the final report and creating graphs.

References

[1] Benton J. Anderson and Jesse G. Meyer. Finding the optimal human strategy for wordle using maximum correct letter probabilities and reinforcement learning. *CoRR*, abs/2202.00557, 2022.

[2] Jack Coop. Building a wordle solver with python, 2022.

[3] Nisansa de Silva. Selecting seed words for wordle using character statistics. *CoRR*, abs/2202.03457, 2022.

[4] Arvi Hurskainen. Optimal selection of words for wordle. WorkingPaper 80, SALAMA - Swahili Language Manager, Finland, January 2022.

[5] Team Interview Kickstart. What is wordle and how to create a wordle solver, 2022.

[6] Daniel Lokshtanov and Bernardo Subercaseaux. Wordle is np-hard, 2022.

[7] Noa Lubin. Wordle — what is statistically the best word to start the game with?, 2022.

[8] Tom Neill. Ruining the fun: a wordle auto-solver, 2022.

[9] Johnathan Olson. Optimal wordle solutions, 2022.

[10] Martin B. Short. Winning wordle wisely, 2022.

[11] Owen Yin. Here lies wordle: 2021–2027 (full answer list), 2022.