Wordle Solver

Jihun Kim
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
TN, USA
jkim172@vols.utk.edu

Seoyoung An
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
TN, USA
san5@vols.utk.edu

Jonathan E. Skeen
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
TN, USA
jskeen6@vols.utk.edu

***Abstract*—This project attempts to create a machine learning model that plays and outperforms humans at the popular New York Times web-based word game, Wordle. This project starts by retrieving a dataset containing 5-letter words. A subset of these words is chosen to become our dataset. A Bayesian network was chosen as our machine learning approach since it meshes well with our problem and is also an alternative to deep learning in this scenario. The model's performance was measured with an average score of 4.09. Then, an algorithm was incorporated to provide the model with the best starting word. This algorithm used data from the initial phase, along with data pertaining to the dataset, to provide the best starting word. After this addition, performance was once again measured with an average score of 3.87. We believe this performance is good, but we also expect further improvements to lower the score even more.**

***Index Terms*—Wordle, Bayesian algorithm, machine learning**

I. Introduction

Our project is trying to solve Wordle with the least amount of attempts. Wordle is a game in which the player needs to guess a hidden word. The hidden word changes every day, and a maximum of 6 tries are given to find the answer. Each guess must need to be a valid 5-letter word. The color of the tiles will change when a player submits the word to show how close the player's guess was to the answer. Green tiles mean that the given character is in the word and in the correct spot. Yellow tiles represent that the character is in the word but in the wrong spot, and grey tiles indicate that the letter is not in the word. [1] Figure 1 shows the tutorial of Wordle, and Figure 2 is an example screenshot of the Wordle game.

At first, we set our goal to find the answer within five attempts. However, after training, we realized that our solver worked better than we initially thought. Therefore, we changed our goal to find the hidden word within three to four attempts, considering that most players find the answer on an average of 4 attempts.

To achieve this goal, we chose the Bayesian model as our machine learning approach. We used the Bayesian model because it makes a decision based on observed data or a prior decision. We used possible 5-letter words and the history of Wordle answers as our datasets. A detailed explanation will be included in the Dataset section. Our model was able to reach 3.87 attempts on average to successfully find a target word in the testing set as a result. To improve our work, we will try different algorithms to lower the average attempts.

A. Motivation

To play Wordle, players must carefully select a word to submit, based on the relationship between the color of the tiles submitted before. We thought these conditions were suitable for machine learning; therefore, we wanted to create a model that finds the target word with the fewest attempts.

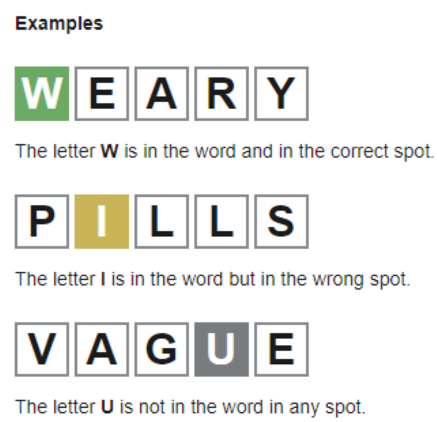

Fig. 1. Wordle Tutorial [1]

Fig. 2. Example Wordle Game [1]

II. Dataset

For our training dataset, we used a preexisting dataset which is a subset of all possible 5-letter words that can be taken for Wordle which were 2,315 words. We used a subset since not all of those 5-letter words are appropriate to be an answer for Wordle. At the beginning of the project, we selected a dataset of all possible 5-letter words that are possible answers for Wordle. However, the running time was too long since the dataset was huge, having 12,953 words. Since some of the 5-letter words were inappropriate to be an answer, the developers of Wordle filtered some of the words. We used previous Wordle answers for the testing dataset, a total of 868 words.

Before creating a solver, we analyzed the training and testing datasets. Figure 3 and 4 show the frequency of each alphabet in the training and testing dataset. Both datasets have a similar trend, having more vowels and the alphabet ’R’, ’S’, and ’T’. ’E’ was the character that occurred the most in the training and testing dataset.

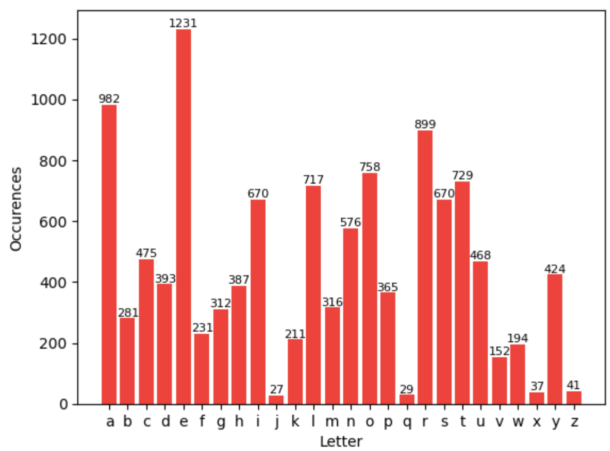

Fig. 3. Letter Frequency - Training

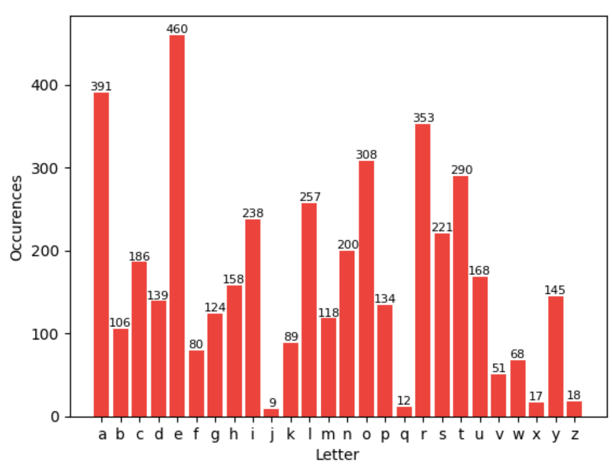

Fig. 4. Letter Frequency - Testing

Based on this analysis, we wanted to find out the ratio between the number of vowels and consonants within the dataset. Figure 5 represents the occurrences of vowels and consonants in the datasets. For the training dataset, the ratio between vowels and consonants was 5.5 to 10. The number of vowels was about a little bit more than half of the number of consonants. For testing, the ratio was 5.6 to 10, very similar to the training dataset.

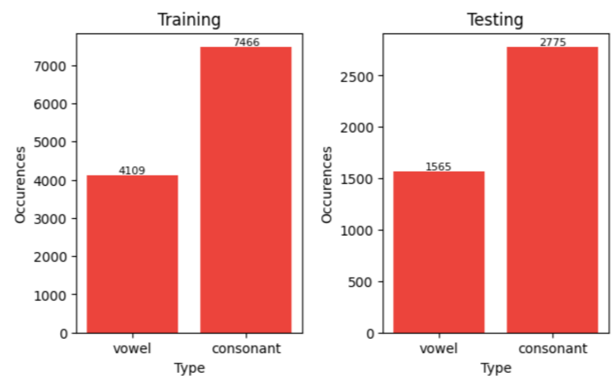

Fig. 5. Letter Type Frequency

Since vowels have a very important role in Wordle, we wanted to analyze how many vowels are in each 5-letter word in the datasets. To do so, we created pie charts to see the number of vowels in every 5-letter word in both training and testing datasets. 0 represents that there was no vowel in the word, 1 means that there was only one vowel in the word, and so on. Both datasets have similar patterns of having one or two vowels in 5-letter words, which is shown in Figure 8 and 9.

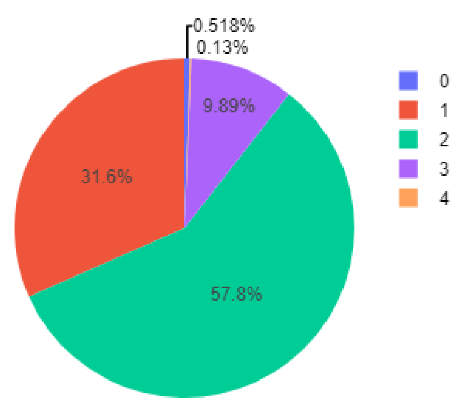

Fig. 8. Number of Vowels in Words - Training

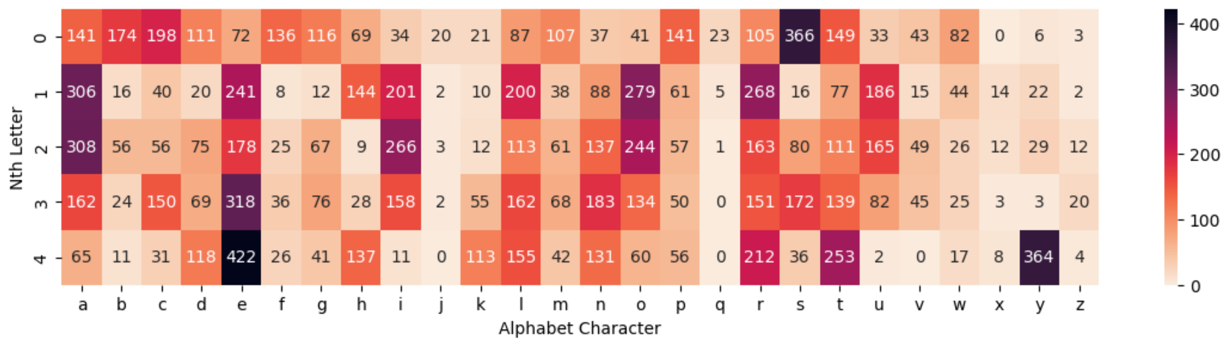

Fig. 6. Letter Frequency Heatmap - Training

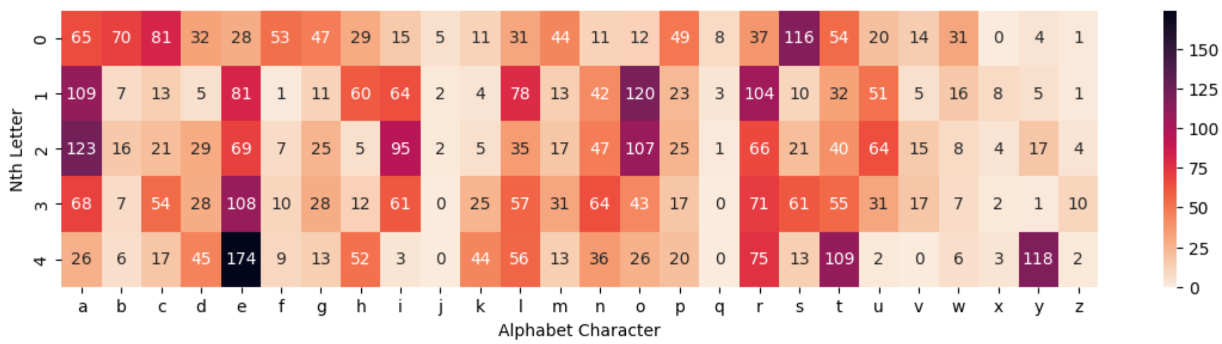

Fig. 7. Letter Frequency Heatmap - Testing

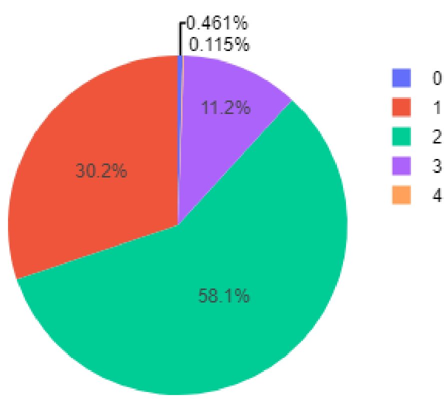

Fig. 9. Number of Vowels in Words - Testing

For the overall distribution of alphabet characters and their locations in the 5-letter words. We visualized it through heatmaps as shown in Figure 6 and 7, where the x-axis indicates each alphabet character and the y-axis represents the location of that character. The color bar shows how many characters are in that location. An interesting finding is that in both datasets, the alphabet 'E' is the most frequent character as the last character of the words. 'S' occurred as the first character the most compared to the other alphabet characters.

III. Approaches and Methodology

A. Machine Learning Approach

For our machine learning approach, many options were not available to us due to the characteristics of Wordle. Instead of having data that needed to be classified or clustered, we needed an iterative algorithm that updates its output based on new information it has received. The first thought involved deep learning, such as building a Recurrent Neural Network (RNN), however, it was not a good choice since it could take some time to find the good next word and recommend it to the user. Our group consulted with Dr. Schuman to get advice on what machine learning algorithm to use for our model. She recommended using a Bayesian for our model.

A Bayesian model involves assigning probabilities to events based on new information it has received. More specific to this scenario, our Bayesian model receives Wordle guesses, as well as the Wordle feedback from those guesses. With these pieces of information, our model assigns probabilities to potential words and returns its best choice for the next Wordle guess.

B. Experimental Setup

The purpose of this project is to determine if machine learning can outperform a human at playing Wordle. Thus our main metric for comparison is human performance at the game. The United States' average Wordle score is 3.92. The country with the highest average score is Sweden, with a score of 3.72 [2]. However, we would like to set our baseline as the world's average score. Thus, if our model can outperform

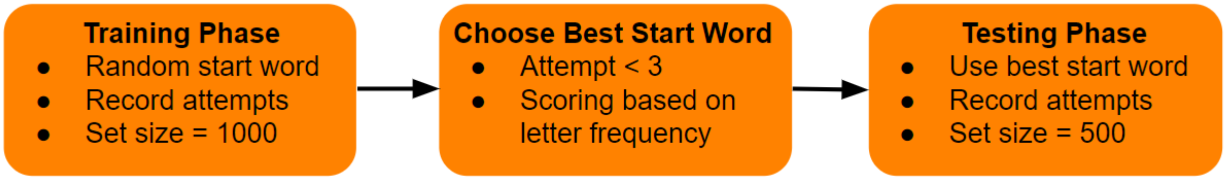

Fig. 10. Bayesian Model Workflow

the world's average of 4.016, then we believe our model is successful.

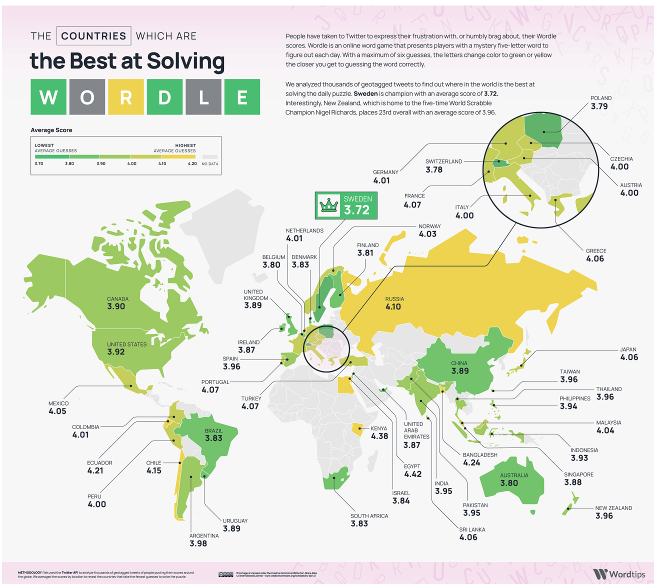

Fig. 11. World's Wordle Scores [2]

C. Build a Bayesian Model

```
word #1: dowel
attempt: 4
word #2: wound
attempt: 4
word #3: shalt
attempt: 2
word #4: forgo
attempt: 5
word #5: milky
attempt: 4
```

Fig. 12. Part of Process of Choosing Target Word and Record Attempts

Bayesian Models don't have hyperparameters. In place of this, our team chose to perform the best first-word optimization. If the goal is minimizing the number of Wordle guesses, then having an informative and accurate first guess is crucial to achieving that goal. We assign probabilities and scoring based on guess accuracy using the feedback and make another guess based on probabilities. If many words have the same score, it chooses a random word from them.

In the training phase, we chose a target word randomly from the training set. We record how many attempts took to find the target word and what is the corresponding start word in each game as fig 12. We repeated this 1000 times so that we could find the best starting word.

```
Word Scores:
earth: 42.006
shone: 39.379
avert: 36.89
rouse: 35.281
singe: 34.759
coast: 33.839
awoke: 32.507999999999996
swept: 32.373999999999995
shout: 31.741999999999997
plier: 31.609
wafer: 31.444000000000003
piano: 31.317999999999998
plead: 31.076
roach: 30.537000000000003
forge: 30.439
baker: 30.119999999999997
hyper: 28.686
turbo: 26.800000000000004
juice: 25.461
dutch: 24.943
bingo: 24.729
smack: 21.453999999999997
corer: 5.029999999999996
villa: 4.036
...
bleed: -28.336
geese: -80.572

Best Word for the First Word in Wordle: earth
```

Fig. 13. Scoring letter frequency

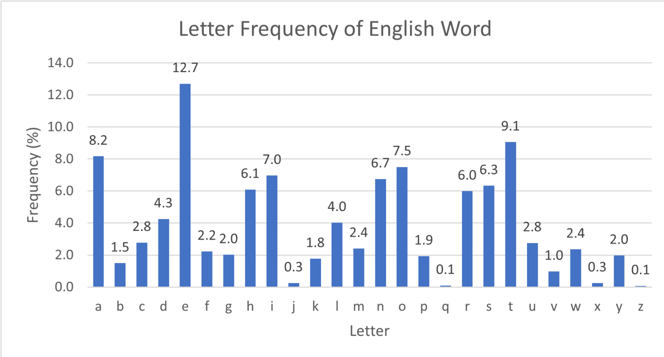

Fig. 14. Letter Frequency Graph [3]

Before we move on to the testing phase, we wanted to choose the best-starting word to make the performance of the Bayesian better. Hence, we selected the start word with less than three attempts and scored each word based on the letter frequency. To make the start word finding a good letter easier, we gave a penalty when a word has the same letter. In multiple runs of the model, "lathe, alert, and irate" were frequently chosen as the best start words.

We set the best start word as a start word and randomly chose a target word from the testing set. Other than that, the process is as same as the training phase.

IV. Results

A. Find the target word

During the training and testing phases, we recorded how many attempts our model using Bayesian needed to find the target word. Since the testing phase uses the best starting word found through the training phase and letter frequency, it shows slightly better results.

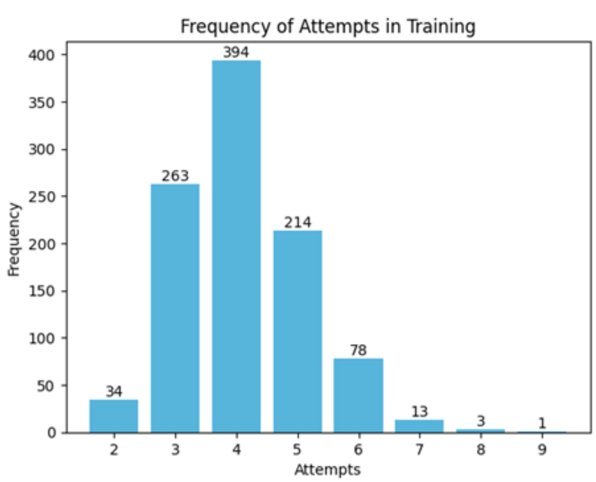

Fig. 15. Frequency of Attempts in Training

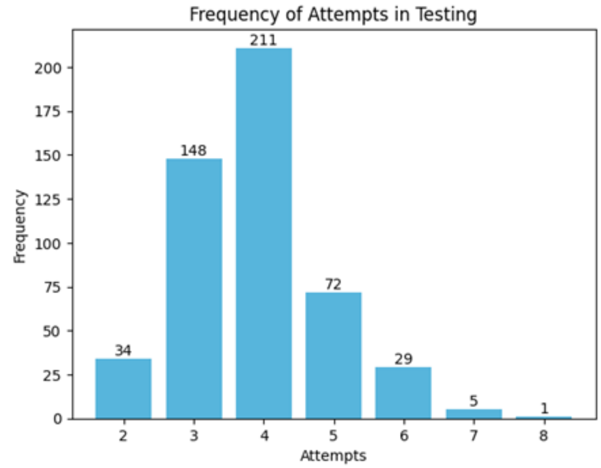

Fig. 16. Frequency of Attempts in Testing

Fig 15 shows the attempts taken by our Bayesian model to find the target word using the training set. Our Bayesian model took two attempts as the minimum attempts, and it took nine attempts as the maximum attempts to find the target word using the training set. 34 words out of 1000 words (3.4%) were found in two attempts. 263 words out of 1000 words (26.3%) were found in three attempts. 394 words out of 1000 words (39.4%) were found in four attempts, and it was the most frequently appeared. 214 words out of 1000 words (21.4%) were found in five attempts. 78 words out of 1000 words (7.8%) were found in six attempts. 17 words out of 1000 words (1.7%) were found in more than seven attempts which means it failed to find words based on the Wordle rule. Using the training set, our Bayesian model succeeded in finding the target word at the rate of 98.3% and it took 4.09 attempts on average to find a target word.

Fig 16 shows the attempts taken by our Bayesian model to find the target word using the testing set. Our Bayesian model took two attempts as the minimum attempts, and it took eight attempts as the maximum attempts to find the target word using the testing set. 34 words out of the 500 words (6.8%) were found in two attempts. 148 words out of 500 words (29.6%) were found in three attempts. 211 words out of 500 words (42.2%) were found in four attempts and it was the most frequently appeared. 72 words out of 500 words (14.4%) were found in five attempts. 29 words out of 500 words (5.8%) were found in six attempts. 6 words out of 500 words (1.2%) were found in more than seven attempts which means it failed to find words based on the Wordle rule. Using the testing set, our Bayesian model succeeded in finding the target word at the rate of 98.8% and it took 3.87 attempts on average to find a target word.

We concluded that the best starting word, which was chosen by training phase and letter frequency helped to improve the performance of the Bayesian model. Furthermore, according to data collected from Wordle players, the global average for solving Wordle is 4.016, and 33.10% of games found the target word in four attempts [4]. Since the average attempts to find the target word using our Bayesian model are fewer than the global average attempts, we concluded that the performance of our Bayesian model is as good as most players globally.

B. Playable Wordle-Solver

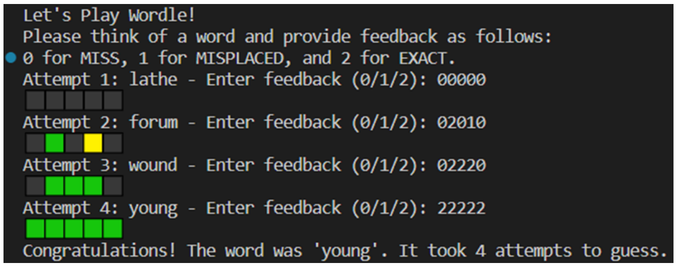

Fig. 17. Playable Wordle-Solver

We used all possible words and letter frequency to find the best starting word, and through previous Wordle answers, we were able to check the performance of our model. We wanted to make this Wordle-Solver usable for playing actual Wordle. To make this possible, we built a system where our model recommends words in real-time, receives feedback on words from users, and then the model predicts the next word through a Bayesian algorithm and informs the user. For users who find it difficult to input colored tiles, we decided to use numbers as a means of sending feedback to our model. 0 represents gray tile, 1 represents yellow tile, and 2 represents green tile. The model is designed to reject any ungrammatical input and will require re-entry. Additionally, feedback sent by users is visualized using color tiles. Fig 17 shows how our Bayesian model and user can communicate with each other, and it shows the Bayesian model could successfully find the word "young" in four attempts using the feedback from the user.

V. Discussion, Conclusion, and Future Work

A. Discussion and Conclusion

The Bayesian algorithm showed excellent performance in selecting words that meet conditions by considering combinations between letters in words. In addition, by using only words that could be the correct answer, we were able to reduce the program execution time and reduce the number of attempts. Additionally, by selecting the best-starting words through letter frequency, our model was able to find the target word in fewer attempts. In conclusion, our model succeeded in finding the target word in fewer trials than the international average. Although our model produces fewer average attempts than the global average of 4.016, it falls short of the lowest average attempts in Sweden at 3.72. We want our model to be a model with superior performance than what humans can do. Therefore, in future work, our new research question would be: Is it possible to lower the average number of attempts to 3?

B. Reduce Attempts: Improvement of Bayesian

We will look for ways to improve the Bayesian algorithm to reduce the number of attempts of our model. To do this, it is necessary to identify the characteristics of words that have an Attempt count of 5 or more. What we understood was that words with a high number of attempts had many words with similar forms in common. For example, to find the target word "sleep," our model might go through words like "sheep" or "sweep". However, if a real person were to play Wordle, they would be able to guess the most likely word out of the three just by typing in a word like "whole". However, our model is designed to randomly recommend one of several words with a similar form. Although we tried using letter frequency to output the word with the highest score, we did not find any significant improvement. Considering this, major modifications to the algorithm may be required to address these issues.

C. Reduce Attempts: Use of different algorithms

If it is difficult to improve these issues using only the Bayesian algorithm, we will also consider using other machine learning algorithms. Combining machine learning algorithms such as decision trees will be of great help in considering a variety of cases at once. For example, if there are several words to recommend, it will be able to select the word that leaves the fewest possible next words if the word is not the correct answer. It will be greatly helpful to choose a more appropriate word as the next word.

We could also consider implementing a deep learning approach. Deep networks have increasingly been applied to various problem-solving tasks and we believe the same networks could thrive in this scenario as well.

Acknowledgment

We would like to appreciate to our professor Dr. Schuman for her insightful guidance and advice.

References

[1] J. Wardle, "Wordle How To Play", https://www.nytimes.com/games/wordle, Oct. 2023 (accessed Dec 8, 2023).
[2] M.Lancu, "Where in the World Is the Best at Solving Wordle?", https://word.tips/wordle-wizards/, Aug. 2023 (accessed Dec 8, 2023).
[3] G. Grigas "Letter Frequency Analysis of Languages Using Latin Alphabet," *International Linguistics Research*, Mar 2018.
[4] M.Broz, "Wordle Statistics, Facts, & Strategies (2023)", https://photutorial.com/wordle-statistics/, Mar. 2023 (accessed Dec 8, 2023).