Wordle 1

Anjnesh Sharma
School of Computing and Information
University of Pittsburgh
Pittsburgh, PA, USA
ans458@pitt.edu

Yunpeng Guo
School of Computing and Information
University of Pittsburgh
Pittsburgh, PA, USA
yug64@pitt.edu

Renjie Xu
School of Computing and Information
University of Pittsburgh
Pittsburgh, PA, USA
rj.xu@pitt.edu

ABSTRACT

Wordle is an online word game that looks a lot like classic codebreaking, color-coded board games like Mastermind, but it's even easier to play [**Figure 2**]. There are over two million daily players of this game, and more than 15 million tweets about Wordle results from 1/2/2022 to 03/06/2022 on Twitter. Wordle became one of the most popular games around the world.

In this paper, utilizing information retrieval and other data mining techniques, we have developed a system that can predict each day's wordle in the first attempt. We achieve this by performing fundamental steps, (1) scarping player results from twitter and cleaning data; (2) perform word simulations to create an index; and (3) predict the day's wordle by ranking words using three parameters. For accuracy check, we predicted every day's wordle from wordle day 1 to the present day and successfully predicted the word each time in the first attempt.

KEYWORDS

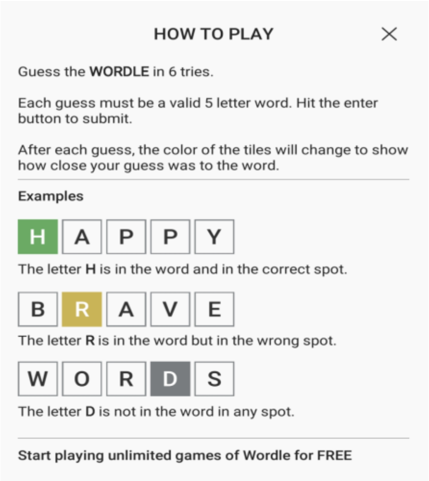
Data mining; Data scraping; Prediction; Indexing; Information retrieval; Twitter; Wordle

1. **Introduction**

Wordle gives players six chances to guess a randomly selected five-letter word. As shown in **Figure 1**, if you have the right letter in the right spot, it shows up green. A correct letter in the wrong spot shows up yellow. A letter that isn't in the word in any spot shows up gray.

At the end of each game, the player can share their result on twitter in the encoded format, shown in **figure 3**. We used this encoded data to predict each day's wordle.

The Wordle source code contains 2,315 days of answers that are all common 5-letter English words and 10,657 other valid, less-common 5-letter English words. We combine these to form a set of 12,972 possible words/answers.



HOW TO PLAY

Guess the **WORDLE** in 6 tries.

Each guess must be a valid 5 letter word. Hit the enter button to submit.

After each guess, the color of the tiles will change to show how close your guess was to the word.

Examples

H A P P Y

The letter **H** is in the word and in the correct spot.

B R A V E

The letter **R** is in the word but in the wrong spot.

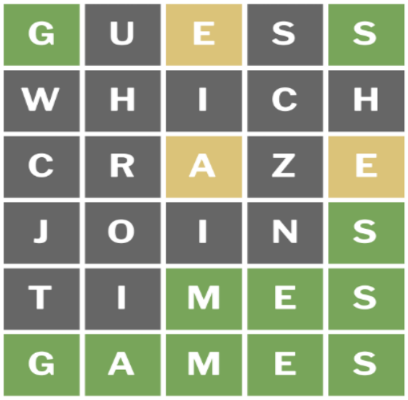
W O R D S

The letter **D** is not in the word in any spot.

Start playing unlimited games of Wordle for FREE

Figure 1: How to play wordle

We then simulate playing 3000 Wordle games for each of these possible words, guessing based on the frequency of the word in the English language and the feedback received. We take three measures to evaluate the observed distribution of ⬛ 🟨 🟩 squares on Twitter according to our valid words. We then generate our guess by taking the word with the best average rank across these three measures.

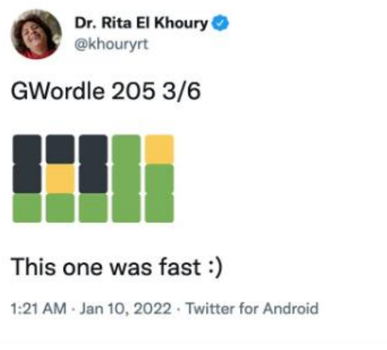


G U E S S
W H I C H
C R A Z E
J O I N S
T I M E S
G A M E S

Figure 2: Sample wordle game

2.1 Data scraping

To predict the day's wordle, we scrape 3000 wordle tweets every day. An example of a tweet is shown below.



Dr. Rita El Khoury
@khouryrt

GWordle 205 3/6

⬛⬛⬛🟩🟨
⬛🟨⬛🟩🟩
🟩🟩🟩🟩🟩

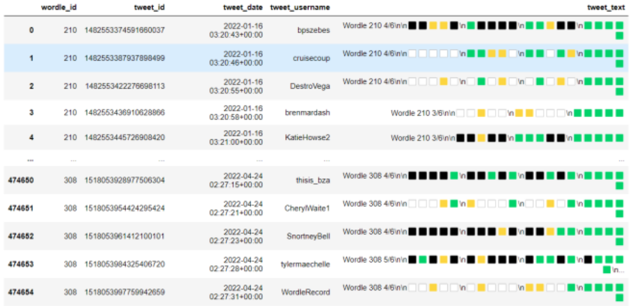
This one was fast :)

1:21 AM · Jan 10, 2022 · Twitter for Android

Figure 3: Wordle tweet for day 205

We used the tweepy library in python to make an api that pulls user tweets. A twitter developers account with elevated access is also needed to get the permission to access these tweets and its contents for academic research purposes.

We identify relevant tweets for the day checking the wordle number. The first wordle was published on June 19, 2021.Counting the number of days after that can help us find the days wordle number.



	wordle_id	tweet_id	tweet_date	tweet_username	tweet_text
0	210	1482553374591660037	2022-01-16 03:20:43+00:00	bpszebes	Wordle 210 4/6\n\n⬛⬛🟨🟨⬛\n🟩⬛⬛⬛⬛\n🟩🟩🟨⬛⬛\n🟩🟩🟩🟩🟩
1	210	1482553387937898499	2022-01-16 03:20:46+00:00	cruisecoup	Wordle 210 4/6\n\n⬜⬜⬜⬜⬜\n🟩🟩🟨🟨⬜\n🟩🟩⬜🟩🟨\n🟩🟩🟩🟩🟩
2	210	1482553422276698113	2022-01-16 03:20:55+00:00	DestroVega	Wordle 210 4/6\n\n⬜⬜⬜🟨⬜\n⬜🟩⬜🟨⬜\n⬜🟩⬜🟨⬜\n🟩🟩🟩🟩🟩
3	210	1482553436910628866	2022-01-16 03:20:58+00:00	brenmardash	Wordle 210 3/6\n\n⬜⬜🟨⬜⬜\n🟨🟨⬜⬜⬜\n🟩🟩🟩🟩🟩
4	210	1482553445726908420	2022-01-16 03:21:00+00:00	KatieHowse2	Wordle 210 3/6\n\n⬛⬛🟨⬛⬛\n🟩🟩🟩⬛⬛\n🟩🟩🟩🟩🟩
...	...	...	...	...	...
474650	308	1518053928977506304	2022-04-24 02:27:15+00:00	thisis_bza	Wordle 308 4/6\n\n⬛⬛⬛⬛🟩\n⬛⬛🟩⬛🟩\n⬛⬛🟩⬛🟩\n🟩🟩🟩🟩🟩
474651	308	1518053954424295424	2022-04-24 02:27:21+00:00	CherylWaite1	Wordle 308 4/6\n\n⬜⬜⬜🟨🟩\n🟨⬜⬜⬜🟩\n⬜⬜🟨⬜🟩\n🟩🟩🟩🟩🟩
474652	308	1518053961412100101	2022-04-24 02:27:23+00:00	SnortneyBell	Wordle 308 4/6\n\n⬛⬛⬛⬛⬛\n⬛⬛⬛🟨⬛\n⬛⬛🟨🟩🟩\n🟩🟩🟩🟩🟩
474653	308	1518053984325406720	2022-04-24 02:27:28+00:00	tylermaechelle	Wordle 308 5/6\n\n⬛🟩⬛🟨⬛\n⬛🟨⬛⬛⬛\n⬛🟩🟩⬛🟩\n⬛🟩🟩🟩🟩\n...
474654	308	1518053997759942659	2022-04-24 02:27:31+00:00	WordleRecord	Wordle 308 4/6\n\n⬜⬜🟨⬜⬜\n⬜🟨⬜⬜⬜\n🟨🟨⬜⬜🟩\n🟩🟩🟩🟩🟩

Figure 4: Scraped twitter data

2.2 Preprocessing

The scraped raw data containing the user tweets that contained a large amount of irrelevant information such as users' comments, the difficulty level of the day's Wordle quest, or feelings of solving the puzzle, but the only useful part for running the simulations was the feedback of each attempt that was in the format of 5 colored squares representing whether the letter at a specific position matched today's answer.

Therefore, data preprocessing was required to filter this part from the raw data. For the data with each wordle_id, the colored squares were encoded as upper-case letters to facilitate understanding, i.e., using 'Y' (yes) to replace the green square, 'M' (maybe) to replace the yellow square, and 'N' (no) to replace the black one. After that, strings composed of these three letters were filtered by regular expressions, and the cleaned data was encoded in a dataframe with columns 'wordle_id', 'tweet_id', 'tweet_date', 'tweet_username' and 'tweet_text', where the 'tweet_text' contained the cleaned user tweets content formatted as strings of 'YMN's.

3. Word simulation

For every word, there are three possible feedbacks for each letter in a guess. This means there are $3^5 = 243$ possible feedback results. We use a 243-dimensional vector to store these result distributions. Note that note every combination in these 243 possibilities are valid in practice. For example, YYYYM will never be seen because if the first four letters are correctly placed and the fifth is also in the word, it will be correctly placed. But we can ignore these edge cases for our purpose.

We use the English Word Frequency [1] dataset to simulate 3000 Wordle games for each word. This dataset contains a ranked list of most frequently used 5 letter English words. At the end of each simulation, we get a distribution of 5 colored squares which is saved. In a separate, 1 x 243 vector, we save the number of times a distribution appears for each word.

This process is similar to the indexing process of information retrieval systems. This action needs to be performed only once to get the index. The results are saved in a pickle file and can be used for all future predictions.

4. Prediction

The precomputed simulated results are now compared to the day's distribution of Wordle tweets to accurately guess the day's answer.

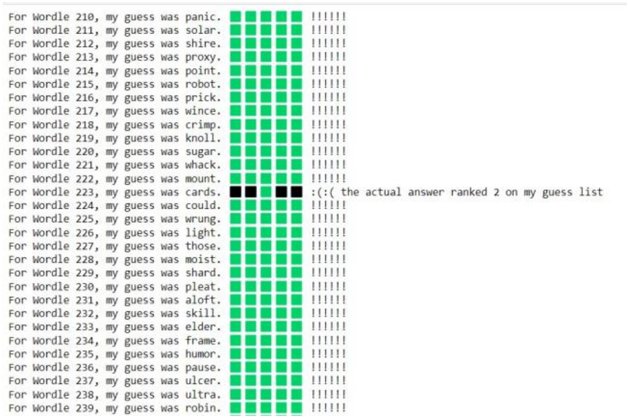
We create another 243-dimensional vector to store the distribution of colored squares and a vector to store the frequency of each pattern. Then we use three measures to evaluate the observed distribution of ⬛ 🟨 🟩 squares on Twitter according to our valid words and rank them.

The first is the frequency of each of the 243 possible 5-square combinations in the observed/simulated games. We rank all the valid words by cosine similarity between the simulated and observed distributions.

The second measure looks at the fraction of these 5 squared combinations that occur right before the correct guess. Again we rank this based on the cosine similarity between the twitter and simulated distributions.

Finally, based on the valid words there's a invalid 5-square combinations for each possible answer. We rank valid words on the number of these invalid combinations we observe. Twitter data being noisy, there's usually some invalid combinations for the correct answer.

We then generate our guess by taking the word with the best average rank across these three measures.



```
For Wordle 210, my guess was panic. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 211, my guess was solar. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 212, my guess was shire. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 213, my guess was proxy. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 214, my guess was point. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 215, my guess was robot. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 216, my guess was prick. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 217, my guess was wince. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 218, my guess was crimp. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 219, my guess was knoll. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 220, my guess was sugar. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 221, my guess was whack. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 222, my guess was mount. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 223, my guess was cards. ⬛⬛🟩⬛⬛ :(:( the actual answer ranked 2 on my guess list
For Wordle 224, my guess was could. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 225, my guess was wrung. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 226, my guess was light. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 227, my guess was those. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 228, my guess was moist. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 229, my guess was shard. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 230, my guess was pleat. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 231, my guess was aloft. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 232, my guess was skill. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 233, my guess was elder. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 234, my guess was frame. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 235, my guess was humor. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 236, my guess was pause. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 237, my guess was ulcer. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 238, my guess was ultra. 🟩🟩🟩🟩🟩 !!!!!!
For Wordle 239, my guess was robin. 🟩🟩🟩🟩🟩 !!!!!!
```

Figure 5: Output

Since we started collecting Twitter data (Wordle 210), this method generated the correct Wordle answer on the first attempt every day except one. The correct word for that day was rank two word in our list.

CONCLUSION AND FUTURE WORK

Based on our comparison rank lists, we generate our guess for hundred different wordle games, and we only get one time that doesn't predict correct answer at first attempt. Thus, our accuracy to predict the right answer is about 99%.

Our current model tries to predict the word in the first attempt and there can be times when it fails. In our future work we plan to combine a probabilistic approach [2], that combines our ranked list with the feedback received in first attempt.

ACKNOWLEDGMENTS

We would like to express our gratitude to Dr. Yuru Lin for her teaching of the data mining processes that inspired us to dig knowledge from the user tweets data and apply them to solve the world puzzle. Also, we would like to thank twitter for providing powerful api to acquire the data we need in a much easier way.

WORK DISTRIBUTION

In our project, Wordle 1, Yunpeng implemented a Python API for data collection (including tweets and frequent words). Besides, he helped the documenting in our project. And he offered many useful literature references at the beginning to help team configure our model.

Anjnesh designed and implemented our word simulation model. He proposed the idea of creating an index, using frequency of each distribution, which we used in our prediction model.

Renjie designed the prediction model. He developed the three ranking metrics and used them to compare the simulated distribution with the twitter distribution. Using this we created a ranked list that was used to guess the word.

This is just a broad categorization of work. All of us have contributed to the data pre-processing, programing and to the final report of our project

REFERENCES

[1] English Word Frequency https://www.kaggle.com/rtatman/english-word-frequency

[2] Wordle Puzzle Solving https://www.kaggle.com/code/uniquekale/wordle-puzzle-solving/notebook