Modeling Challenge III: Ghost

Eckel, TJHSST AI2, Spring 2022

Background & Explanation

We’ve been working on turn based game AI; this is an assignment to model a very different game and figure out how to win.

The game is Ghost and it goes like this.

- Players take turns each writing a letter.
- Every time a letter is written, the sequence of letters so far must be the beginning of at least one valid word.
- If you are the player that *finishes* the word, you *lose.*
- Short words don’t count and what “short” means can vary. I’ll specify the minimum length as an argument.

Say, for example, that the game so far is:

ABSEN

…and it’s your turn. You can’t play “Q”, because “ABSENQ” is not the beginning of a valid word in English. You could play “T”, making “ABSENT”, but then you’d lose. (Note that this is true even though the word could, in theory, continue; “ABSENTLY” is even longer. But you lose as soon as you write the last letter of a word, even if there are longer words that start that way.) So, probably you want to play “C”, making the game so far:

ABSENC

…because then the only letter your opponent can play is “E”. They finish the word “ABSENCE”, they lose, you win.

Required Task

As with all modeling challenges, I’m not giving you any information on how to do this. Use anything you’ve learned so far that seems helpful. Your job is to take a text file containing valid words, a minimum required word length, and a game in progress (ie, partially completed word) and to return a list of all choices of next letter that will GUARANTEE that the next player will ultimately win the game. Note this does NOT mean that the opponent’s NEXT MOVE will end the game; it means that no matter what the next move is, the game will EVENTUALLY end up as a victory for the current player!

IMPORTANT NOTE: As you read in the dictionary, be sure to discard any words that are too short or any words which contain non-alphabetic characters (ie, digits or apostrophes, etc). The python function `isalpha()` may be of use.

Unlike modeling challenge I, **you cannot declare failure on this assignment**. You can work with a partner. How you work with your partner is entirely up to you; I will not regulate, require, or judge any aspect of your teamwork. That’s up to you. You may also work alone if you prefer. You can break up with your partner at any time; just tell me.

Specification for RED Credit

Submit a single Python script to the link on the course website.

This assignment is **complete** if:

- You follow the instructions on the submission form to format your submission properly. If you're working with a partner, make sure you at least include both last names and first initials (like "5 Galanos R Eckel M").
- Your code does all of the following:
    - Accept the following command line arguments:
        - First, the name of a text file containing the dictionary to be used. **This may contain upper case or lower case words – be sure to standardize your input! (Also read the NOTE above.)**
        - Second, the minimum length of word that will be allowed (a single integer).
        - Third, ***optionally***, a game in progress. If no third argument is present, start from the first move. No matter what, assume that the player your AI is representing plays next.
    - Output a list of letters that the next player can use to guarantee victory. If no victory can be guaranteed, output "Next player will lose!"
- Total runtime is less than 2 minutes.

Examples

This is the output of several runs of my code. These all use the words_all.txt file linked on the course website. Each run prints out an answer in under 5 seconds. You'll have to do several tests like this in less than 2 minutes total on a similarly-sized dictionary.

```
>ghost_example_solution.py words_all.txt 4
Next player can guarantee victory by playing any of these letters:  ['M', 'N']

>ghost_example_solution.py words_all.txt 4 A
Next player can guarantee victory by playing any of these letters:  ['Q', 'V']

>ghost_example_solution.py words_all.txt 4 AB
Next player can guarantee victory by playing any of these letters:  ['J', 'V', 'W', 'Y']

>ghost_example_solution.py words_all.txt 4 ABY
Next player will lose!

>ghost_example_solution.py words_all.txt 3
Next player will lose!

>ghost_example_solution.py words_all.txt 3 A
Next player can guarantee victory by playing any of these letters:  ['A', 'Q', 'V']

>ghost_example_solution.py words_all.txt 3 AB
Next player can guarantee victory by playing any of these letters:  ['J', 'V', 'W']

>ghost_example_solution.py words_all.txt 3 ABE
Next player can guarantee victory by playing any of these letters:  ['A', 'C', 'R']

>ghost_example_solution.py words_all.txt 3 ABEC
Next player will lose!
```