CS 4300 - WordGuess PEAS Assessment - Yousof Abusamra

The game of Wordle has a system for deducing a hidden five-letter word through strategic guessing. The guessing system uses letter placement and feedback to advance toward the solution. Feedback can only be obtained by submitting valid five-letter words as guesses, with each guess revealing which letters are correct (green), present but misplaced (yellow), or absent (gray). A useful AI agent would help plan which words to guess based on the accumulated feedback, optimizing the information gained from each attempt. The optimal result would be identifying the target word in the minimal number of guesses, given constraints on the valid word list and the six-guess limit.

The Performance Measure:

The following elements will be added together and then the negative of the sum will be used to produce the performance measure. The best agent will have the greatest performance measure. Note that the perfect score is 1, which is unobtainable, unless the target word is guessed on the first try.

- `1*(number of guesses used to find the word)`
- `10*(number of letter locations that reveal no new information)`
- `20*(number of guesses that reveal no new information)`
- `+1000 Penalty if word is not guessed in the six guesses limit`

Additionally the following measure is used in the search process to best guide towards a solution. But because it is too large and unpredictable it would not provide the user with much information if used in the performance measure.

- `1*(number of possible words given the available knowledge)`

The Environment is:

- Observability: Partially observable. The puzzle starts with zero information given about the words other than the letter count. And information is obtained through exhausting guesses.
- Uncertainty: Stochastic. The actions are deterministic. However, the result of the actions and which guesses will reveal new information is unknown until a guess is submitted.
- Duration: Episodic. Each word is a separate game, and resets all counters.
- Stability: The game is static
- Granularity: Discrete. All values are discrete. All values are incremented in whole numbers
- Participants: Single-agent.
- Knowledge: Known. The game system is known and the progression and duration of the game is known, the word is unknown but that falls under observability.

The Actions (Actuators):

- Submit a valid 5 letter string as a guess. This will provide feedback on letter correctness and consume one of the six available guesses.

The Percepts (Sensors):

- A 5 letter string denoting the feedback

  e.g. “ccwae”

  - c → Correct letter and in correct position
  - w → Letter exists but in wrong position
  - a → Letter is absent
  - e → empty feedback, no guess made yet
- Additionally the list of valid words used to select the hidden word is made available through a static method in the model

Search Algorithm Used:

The search algorithm used in the WordGuess project is primarily a heuristic-based approach, which can be classified as an informed search strategy. This method combines elements of constraint satisfaction and probabilistic reasoning to efficiently navigate the large search space of possible words.

The agent utilizes the list of possible words used in drawing the hidden word, and progressively filters based on feedback from previous guesses. This filtering process was shown through search results to always reach the solution when unconstrained.

However, to find the hidden word efficiently and within the allowed six guesses, the agent evaluates potential word choices and scores them according to two main factors:

Letter frequency: Prioritizing words containing frequently occurring letters in the remaining word list.
Letter uniqueness: Favoring words with a higher proportion of unique letters.
By finding a balance between these factors, the algorithm attempts to maximize information gain with each guess, aiming to eliminate as many possibilities as quickly as possible.

While not guaranteed to find the optimal solution in all cases, this heuristic search approach provides a good balance between computational efficiency and solution quality for the WordGuess problem.

Problem Instances:

There is no single consensus on how many 5 letter words there are. Some dictionaries list more than 150,000 while others put the number at 9,000. I have provided the environment with three difficulties that scale by the size of the word list used.

- 3103 words
- 5757 words
- 14855 words

Many of the words used in the latter two lists are rarely (if at all) used by English speakers, but it wouldn't matter to a trained agent.

Now in my search results I have made the word list available to the agent, but a problem instance would be to work with no knowledge of what words were used. This is unforgiving as even a human player has some sort of idea what words might be used. But it is an interesting take on the problem.

Another problem instance would be to play around with the number of guesses permitted, but it goes without saying if you had infinite guesses, you will stumble across the right answer eventually.

Search Results:

The agent is capped at Six guesses only, after which the episode is considered a failure. The environment supports Three levels of difficulties scaling with the size of the word list used.

Following is the performance of the agent over 1000 episodes:

	List of 3k words	List of 5k words	List of 14k words
Success rate (%)	95.8%	90.5%	84.1%
Average number of guesses **left**	2.07	1.84	1.49

To measure the agent's performance if it would have reached a correct solution without a guess limit. Here is the performance of the agent over a 1000 episodes and uncapped guesses:

	List of 3k words	List of 5k words	List of 14k words
Success rate (%)	100%	100%	100%
Average number of guesses **used**	4.09	4.47	5.03

I believe this means with few performance adjustments it is possible to have near perfect results across all lists. The filtering side of the agent is what ensures a correct result is reached eventually, it is the heuristic side of the agent that will need additional work.

Conclusions:

The WordGuess project was my way of exploring artificial intelligence and gymnasium as a whole. The game of wordle, designed for human players, is a classic playground for search strategies. The structure of the environment and the modularity of design provide a good foundation for experimenting with different AI and machine learning techniques.

The project underscores the potential of heuristic-based search strategies in solving complex information retrieval problems, and any problems really. By having an insight and deep understanding of a problem it is possible to bypass logic and massively improve performance.