Analysis Report: Hangman Solver using HMM and Reinforcement Learning

1. Key Observations

Thedevelopmentofthe Hangman solver involved multiple interacting components—namely, the Hidden Markov Model (HMM) for probabilistic reasoning over letters, and the Reinforcement Learning (RL) agent for optimal decision-making. The most challenging aspects were tuning the reward function and managing the balance between exploration and exploitation. Another key challenge was designing a meaningful observation vector that integrated both symbolic state information (masked word, guessed letters) and probabilistic information (letter likelihoods from HMM). We also encountered performance constraints during evaluation due to large word corpora and dynamic environment resets. An important insight was that curriculum learning—starting with short words and progressing to longer words—significantly stabilized learning and improved accuracy.

2. Strategies

TheHMM was trained as a character-level sequential model derived from a large English corpus (corpus.txt). Each character was treated as a state transition, allowing the model to capture dependencies such as 'Q' almost always being followed by 'U'. This design allowed the system to estimate conditional probabilities for unseen word positions, which the RL agent could leverage as priors during letter selection. The RL environment followed the Gymnasium API, where the state space consisted of four major components: (1) the masked representation of the current word, (2) the set of guessed letters encoded as a binary vector, (3) the remaining lives and normalized word length, and (4) the probabilistic output vector from the HMM. The reward function was crafted to strongly encourage correct guesses (2000) while penalizing wrong guesses (-5) and repeated guesses (-2). This asymmetric structure ensured that the agent quickly learned to avoid redundant and random guesses.

3. Exploration vs. Exploitation

Anepsilon-greedyexploration strategy was adopted to manage the trade-off between exploration and exploitation. Initially, a high epsilon value ($\varepsilon = 0.9$) encouraged exploration across a wide range of letter choices to avoid local optima. As training progressed, epsilon decayed exponentially ($\varepsilon \leftarrow \varepsilon \times 0.9995$) to promote more deterministic and high-value actions based on the learned Q-values. The inclusion of the HMM probabilities provided a "soft guidance" even during exploration, making random actions statistically informed rather than uniform. This hybrid approach—combining probabilistic priors with learned Q-values—proved effective for balancing curiosity and exploitation of knowledge.

4. Future Improvements

Ifadditional time wereavailable, several enhancements could be introduced to improve robustness and performance. First, replacing the tabular Q-learning framework with a Deep Q-Network (DQN) would allow better generalization over the high-dimensional continuous state space. Second, integrating a Transformer-based or BERT-style language model could refine the HMM by incorporating semantic context instead of purely character-level dependencies. Third, optimizing the evaluation pipeline—through vectorized environment instances or distributed computation—could greatly reduce runtime for large-scale experiments. Lastly, advanced reward shaping or transfer learning from pre-trained word embeddings could accelerate convergence and yield more human-like guessing behavior.

OUTPUT:

--- FINAL SPRINT: Q-LEARNING + CURRICULUM ---

Loading all words for curriculum...

Loaded 49979 train words, 2000 test words.

Initializing HMM Oracle...

Loaded 49979 words for filtering.

No HMM matrices found. Calculating from corpus...

HMM matrices calculated and saved.

HMM probability matrices are ready.

--- Starting Curriculum Training (Total 10000 Episodes) ---

--- STAGE: Short Words (3-5) (3897 words, 2000 episodes) ---

100%|██████████| 2000/2000 [02:01<00:00, 16.52it/s]

Stage Done. Avg Reward: -18.70, Avg Wrong: 3.52

--- STAGE: Medium Words (6-8) (15214 words, 4000 episodes) ---

100%|██████████| 4000/4000 [07:50<00:00, 8.50it/s]

Stage Done. Avg Reward: 78.10, Avg Wrong: 2.08

--- STAGE: Long Words (9+) (30738 words, 4000 episodes) ---

100%|██████████| 4000/4000 [08:42<00:00, 7.66it/s]

Stage Done. Avg Reward: 115.35, Avg Wrong: 1.09

--- Training Complete. Total Time: 1114.43s ---

--- FINAL SPRINT: Q-LEARNING + CURRICULUM ---

Loading all words for curriculum...

Loaded 49979 train words, 2000 test words.

Initializing HMM Oracle...

Loaded 49979 words for filtering.

No HMM matrices found. Calculating from corpus...

HMM matrices calculated and saved.

HMM probability matrices are ready.

--- Starting Curriculum Training (Total 10000 Episodes) ---

--- STAGE: Short Words (3-5) (3897 words, 2000 episodes) ---

100%|██████████| 2000/2000 [02:01<00:00, 16.52it/s]

Stage Done. Avg Reward: -18.70, Avg Wrong: 3.52

--- STAGE: Medium Words (6-8) (15214 words, 4000 episodes) ---

100%|██████████| 4000/4000 [07:50<00:00, 8.50it/s]

Stage Done. Avg Reward: 78.10, Avg Wrong: 2.08

--- STAGE: Long Words (9+) (30738 words, 4000 episodes) ---

100%|██████████| 4000/4000 [08:42<00:00, 7.66it/s]

Stage Done. Avg Reward: 115.35, Avg Wrong: 1.09

--- Training Complete. Total Time: 1114.43s ---

--- Running 2000 Game Evaluation on test.txt ---

100%|██████████| 2000/2000 [04:46<00:00, 6.98it/s]

--- Evaluation Complete ---

--- 📊 Final Hackathon Report ---

Total Games Played: 2000

Total Wins: 512 (25.60%)

Total Wrong Guesses: 9403

Total Repeated Guesses: 0

Success Rate Score: + 512.00

Wrong Guess Penalty: - 47015

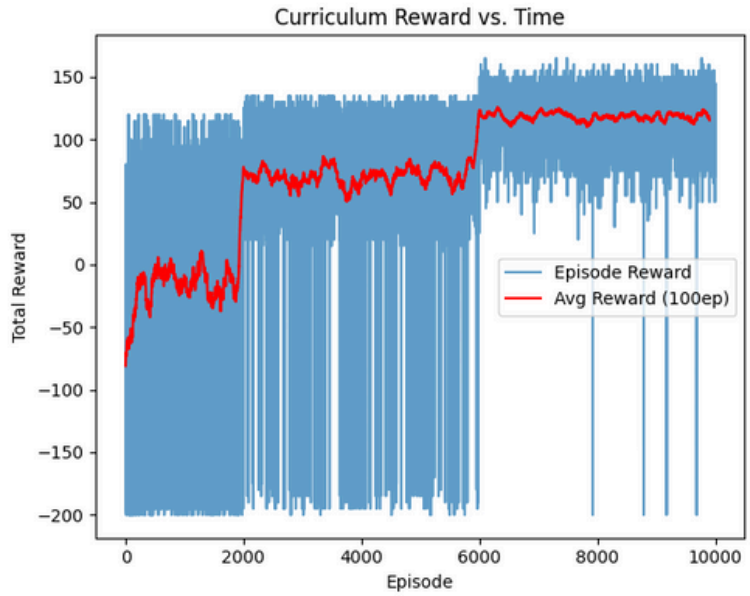
Repeated Guess Penalty: - 0

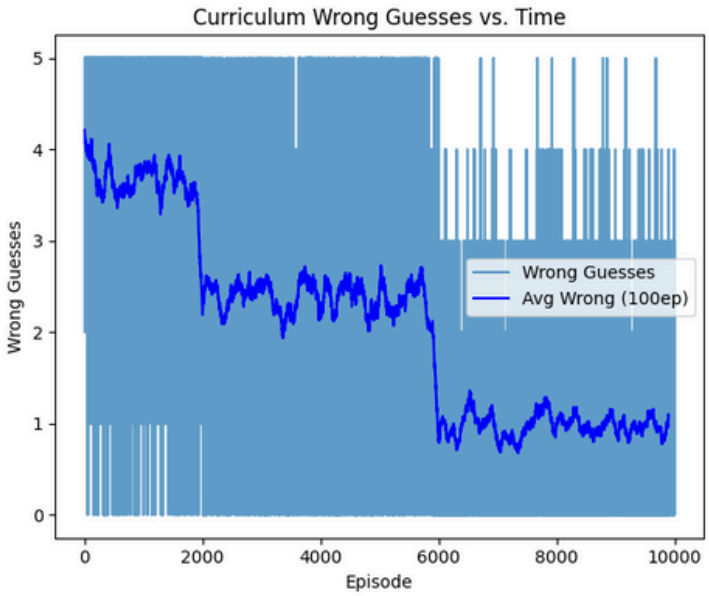
🏆 FINAL SCORE: -46503.00

--- Averages for Analysis_Report.pdf ---

Avg. Wrong Guesses per game: 4.70

Avg. Repeated Guesses per game: 0.00


Curriculum Reward vs. Time
Total Reward
Episode
Episode Reward
Avg Reward (100ep)
150
100
50
0
−50
−100
−150
−200
0
2000
4000
6000
8000
10000


Curriculum Wrong Guesses vs. Time
Wrong Guesses
Episode
Wrong Guesses
Avg Wrong (100ep)
5
4
3
2
1
0
0
2000
4000
6000
8000
10000