

Hangman Solver

Annangi Shashank Babu, EE21B021

July 30, 2023

1 Introduction

The objective of this document is to present an algorithmic strategy designed to play Hangman optimally. Our mission is to devise an intelligent and efficient approach that selects letters for guessing with the highest likelihood of success.

2 Algorithmic Overview

- The algorithm follows a systematic approach, utilizing a scoring mechanism to identify the most promising characters for the next guess.
- The algorithm employs synchronization weighting, penalty consideration, starting value adjustments, and incremental factors to create a sophisticated scoring system that intelligently selects letters for guessing in the Hangman game.

3 Scoring and Matching: Unraveling Word Similarity

- At the start of the algorithm, it aims to make an initial guess for the next letter based on the frequency of letters in the dictionary of potential words until certain percentage of the word is revealed (Found the sweetspot at around 20 percentage). This approach is often referred to as "frequency-based guessing" and provides a reasonable starting point when there is limited information about the target word.
- The sync factor enhances pattern synchronization by considering the alignment of characters in both the guessed word pattern and the candidate dictionary words.
- upon observation and experimentation, it has been noticed that the sync factor plays a pivotal role in the effectiveness of the optimized Hangman-playing algorithm. As a result of its significance, the sync factor is given an exponential weight to amplify its impact on the scoring system.
- The second factor in the optimized Hangman-playing algorithm involves exponential decay to reward matching adjacent letters, elevating the significance of letter sequences and facilitating more refined and strategic letter selections. This thoughtful weighing of letter matches based on proximity provides deeper insights into potential word matches, contributing to more intelligent and accurate guesses as the game unfolds.
- The decay factor is calculated using the formula
$$\text{inc} = \text{pow}(\frac{\text{short_dash}}{\text{len_word}}, 1.1).$$
Here, `short_dash` represents the minimum length of the contiguous sequence of unknown letters (represented by dots) in the guessed word pattern, and `len_word` represents the total length of the guessed word pattern. By applying a power of 1.1 to the ratio of `short_dash` to `len_word`, the algorithm introduces a decay function that rewards adjacent letter matches more than isolated ones. This enhancement allows matching letters in close proximity to each other to receive higher scores, promoting a more nuanced assessment of candidate words and further refining the guessing strategy for the Hangman game.
- The dynamic decay factor allows the algorithm to adjust its scoring approach based on the specific word pattern and length of unknown letters. This adaptability enhances the algorithm's performance across various Hangman puzzles, making it a more versatile and effective strategy.

4 Performance

- The algorithm was put to the test in 1000 games, and it achieved an total of 607 wins, resulting in success rate of approximately 61 percent.
- This success rate is a testament to the algorithm's intelligent and strategic approach to guessing letters. By dynamically adjusting its scoring system, prioritizing adjacent letter matches, and leveraging the significance of the "sync" factor, the algorithm consistently made informed and accurate guesses.
- In conclusion, this endeavor to design an optimal Hangman-playing algorithm has been an enjoyable and rewarding journey. Throughout the process, I found great satisfaction in refining the weightings of various factors and discovering the significance of key parameters.
- Playing with the algorithm was a fascinating experience. Witnessing how the scoring system intelligently weighed letter matches based on proximity and synchronization brought a new level of excitement and challenge to the Hangman game.