MOSAIC PS-2 BABY DRIVER

Aryansh Kumar, Mitul Agarwal, Aditya Raj

HANGMAN SOLVER:

Hangman is a classic word-guessing game typically played by two or more players. One player thinks of a word, phrase, or sentence and draws a number of dashes on a piece of paper, each representing a letter in the word. The other player(s) then attempt to guess the word by suggesting letters, one at a time.

If the guessed letter is present in the word, the player who thought of the word reveals all occurrences of that letter in the appropriate positions on the dashes. If the guessed letter is not in the word, the other player(s) draw one element of a hangman stick figure (such as a head, body, arms, legs, etc.) as a penalty.

The game continues until the guessing player(s) either successfully guess the word or complete the drawing of the hangman, at which point they lose. The word is typically revealed at the end, regardless of whether it was guessed correctly or not.

In Given PS blank word is given by Computer and we need to guess letters and 6 chances are given.

APPROACH:

STEP 1:

Frequency model –

Since in the beginning of the game with very little idea in which direction to go just based on the basis of word length , we choose to fill the initial

10-20% of the word based of the frequency of the occurrence from the given data and also prioritize vowels and made a list with descending frequency of occurrence and prioritizing vowel

STEP 2:

Statistical Window model-

But our initially based model isn't good for generalization as it doesn't relate to the position and relation of letters with other letters . Therefore we have constructed model which takes into account near letters wrt center letter with a window size of 2 like in case of _ _ a _ _ b we would get list of scores of 'a'-'z' for the indices 0,1,3 and 4 wrt 'a' and 3 and 4 wrt 'b' and after that we add up this scores of output the letter with highest scores till 40-50% of the word

STEP 3:

Fined tuned model-

Our second model looks like a good enough fit but it tend to overfit to guess the word which it learnt from the train data in case it already have guessed an average of 4 of letter of the word (it depend of the word size) the scores of top letters might become close to each other due to the summation and hence at later period we use help of model trained by masking training data to 50% by removing random letters and create a corresponding answer set And after we represent each letter in form of vector of size 27 corresponding to all letters and '_' and also pad it to the limit of 27 letters for each word and thus any word (X[0]) can be represented in form of (27,27) array , similar for y i.e. y[0].shape = (26,) , and after making these X and y , we trained it on XGBoost model (here we can also use any other model) and then fine tuned the model with Optuna to get a better fit .

Thus . our final model is the combination of
frequency+statistical_window+fine_tuned_xgboost :
Paste these in a folder named “alphabets_model” and upload it at the same place as the Mosaic folder and change the path according to your drive:)

Future Scope:

As for future:
1. **Reinforcement Learning (RL)**: Integrating RL techniques, especially deep RL or multi-agent RL, could significantly improve the AI's ability to play Hangman. By allowing the AI to learn from its interactions with the game environment, it can adapt its strategies over time, potentially leading to more optimal guessing patterns and better overall performance.

2.**Data Augmentation and Fine-Tuning**: Training the AI on more diverse and extensive datasets can help improve its performance on unseen data. This could involve using larger corpora of words and phrases, including specialized vocabularies or domain-specific terminology. Fine-tuning the model on such data can enhance its ability to guess words accurately in various contexts.

3.**Pretrained Language Models**: Leveraging pretrained language models like GPT-2, LLM2, or successors can be highly beneficial. These models have been trained on vast amounts of text data and have learned rich representations of language. By fine-tuning these models specifically for the task of playing Hangman, the AI could gain a deeper understanding of word patterns, semantics, and contextual clues, ultimately leading to more accurate guesses.

4.**Hybrid Approaches**: Hybridizing Reinforcement Learning with pretrained language models and integrating human feedback loops can enhance

Hangman AI, leveraging diverse methods for improved performance and adaptability.