**Name**- Aarya **Name**- Sneha Pareek

**Entry No**- **Entry No**- 2022CS11096

**An agent designed to correct Automatic Speech Recognition (ASR) errors.**

**Core Idea**

We implemented two functions, sound\_similar and vocab, to perform phoneme-based and vocabulary-based corrections, respectively. The approach utilizes a local search algorithm that moves to the best neighboring state at each step. The vocabulary-based correction is applied after the phoneme-based correction to ensure that the cost function computes the optimal value when inserting words at the beginning or end of the sentence.

**Algorithm Explanation**

* **Reversing the Phoneme Table:** We reverse the phoneme table to easily replace a phoneme with its corresponding key value.
* **Phoneme-based Correction (sound\_similar function):** For each word, we iterate over all possible phonemes of lengths 1 and 2. If a phoneme is present in the phoneme table, we calculate the cost of the new sentence after replacing it. If the new cost is lower than the previous cost, we update both the cost and the sentence.
* **Vocabulary-based Correction (vocab function):** We test all possible combinations of word insertions—either at the beginning, the end, or both. The combination with the lowest cost becomes the final result if it is less than the cost obtained from the sound\_similar function.
* **Iteration Control:** We use a variable max\_iterations to ensure that the sound\_similar function continues to run until no further improvement is found (i.e., the new cost is not lower than the previous cost), indicating that we have reached a global minimum solution.

**Why choose Local Search?**

Local search enabled us to move toward a suboptimal state at each step. We also tried implementing simulated annealing, but it did not improve the solution for this particular agent. The reason was that after reaching the optimal value, simulated annealing could potentially move the solution back to a suboptimal state. Additionally, since the goal state is not fixed, we couldn't set a maximum number of iterations to continue searching. With simulated annealing, further searches after reaching the optimal solution could continue endlessly, whereas local search reliably converges to the optimal solution without risking unnecessary backtracking.