Braille Autocorrect and Suggestion System - Write-up

Overview:

This project implements a browser-based autocorrect and suggestion system for Braille input using QWERTY keyboards. It aids visually impaired users by correcting input errors based on common Braille typing mistakes.

How It Works:

- Users input Braille characters using mapped QWERTY keys: D, W, Q, K, O, P.
- Each Braille letter is formed by pressing a combination of these keys.
- The app uses Levenshtein distance to compare the typed pattern against a dictionary of known Braille words and suggests the closest matches.

Key Features:

- Real-time web interface using Flask.
- Supports common Braille input errors: missing, extra, or mistyped dots.
- Lightweight, fast, and optimized for real-time performance.
- Dictionary and Braille mapping can be extended easily.

Sample Input and Output:

Input: DK W KO

Output Suggestions:

- cat (score 0)
- cab (score 1)
- car (score 1)

Technologies Used:

- Python 3
- Flask Web Framework
- Levenshtein Distance Algorithm
- HTML (via Flask templating)

Trade-offs and Optimizations:

- Precomputed Braille mappings to reduce computation time.
- Limited word dictionary to ensure faster lookup (can be scaled with Trie for large data).
- Stateless design ensures low memory usage for each request.

Future Enhancements:

- Uploadable dictionaries.
- Learning-based suggestions using frequency/rank.
- Support for Grade 2 Braille contractions and multilingual capabilities.

Sample Test Cases

Case 1:

Input: DK W KO

Expected Suggestions: cat, cab, car

Case 2:

Input: DQK W KO

Expected Suggestions: dog, car, cab

Case 3:

Input: DWQK

Expected Suggestions: g, f, d

Case 4:

Input: DWQO DP

Expected Suggestions: bar, bat, cat