Spelling Fixer
Rasika Bhalerao
Part 1

This assignment is to write a spelling fixer using a Hidden Markov model. You will take user input and correct the spelling using the Viterbi algorithm.

Learning goals:

- Calculate emission and transition probabilities for a Hidden Markov model
- Use those probabilities to correct spelling errors using the Viterbi algorithm

What to do:

1. Get a dataset of user text typos. The dataset will need misspelled words which are labeled with the correct version of the word. It would be nice to get a "real world" dataset of "real" typos, but it is also fine to generate a dataset yourself (by taking a dictionary and, for each word, making a few copies of the word with a few randomly selected letters replaced with other letters near it on the keyboard).
    a. Here's one example: https://www.kaggle.com/datasets/bittlingmayer/spelling
2. Write code to calculate the emission probabilities. To do this, iterate through each word character-by-character. For each "correct" letter, calculate the frequency with which each typed letter is emitted. Remember, in most cases, the highest emission probability for each "correct" letter should be the letter itself.
3. Write code to calculate the transition probabilities. Using the "correct" letters, calculate the probabilities for going from the `start` state to each letter, from each letter to each other letter, and from each letter to the `end` state.
4. Write code to correct user text.
    a. Take some user input text, and split it into words using whitespace.
        i. Python's `split()` function is useful.
    b. For each word, use the Viterbi algorithm to decode it.
        i. The typed letters are the emissions. The "correct" letters are the states.
        ii. Print the decoded word.
5. Test it out!

What to turn in:

Please submit these files:

- Your Python code
- A text or pdf file with your answers to these questions:
    - Questions specific to this assignment:

        - Please give an example of a word which was correctly spelled by the user, but which was incorrectly “corrected” by the algorithm. Why did this happen?
        - Please give an example of a word which was incorrectly spelled by the user, but which was still incorrectly “corrected” by the algorithm. Why did this happen?
        - Please give an example of a word which was incorrectly spelled by the user, and was correctly corrected by the algorithm. Why was this one correctly corrected, while the previous two were not?
        - Consider Step 1, which is selecting a dataset of words and their typos. How might the overall algorithm’s performance differ in the “real world” if that training dataset is taken from real typos collected from the internet, versus synthetic typos (programmatically generated)?
    - Questions we ask for every assignment:
        - How long did this assignment take you? (1 sentence)
        - Whom did you work with, and how? (1 sentence each)
            - Discussing the assignment with others is encouraged, as long as you don’t share the code or answers.
        - Which resources did you use? (1 sentence each)
            - For each, please list the URL and a brief description of how it was useful.
        - A few sentences about:
            - What was the most difficult part of the assignment?
            - What was the most rewarding part of the assignment?
            - What did you learn doing the assignment?
        - Constructive and actionable suggestions for improving assignments, office hours, and class time are always welcome.