Maryam Hosseinali - 610398209
homework 1 - question3

3. Is it reasonable to use the difference of summation of ASCII codes of letters for two words as the distance instead of Levenshtein distance? Why?

The idea may initially seem appealing because of its simplicity. However, it is not a reasonable alternative to the Levenshtein distance, for several important reasons:
(So, no! It is not reasonable)

➔ It Ignores Character Position:

One of the main reasons Levenshtein distance is effective is because it considers the position of characters in a word. The Levenshtein distance takes into account the number of insertions, deletions, and substitutions needed to transform one word into another. This makes it sensitive to both the characters and their positions within the word.

In contrast, using the summation of ASCII codes does not account for the order of characters. For example, the words "abc" and "cab" would have the same sum of ASCII values, even though their order is completely different. This means that using ASCII sums as a distance measure would treat "abc" and "cab" as identical, even though their Levenshtein distance is 2 (since two transpositions are needed to convert one to the other).

➔ Different Words Can Have Similar ASCII Sums

Another problem with using ASCII sums is that different words can have similar or even identical sums, despite being very different. For instance, the ASCII sum of the word "cat" is 312, while the sum for "dog" is 314. The difference is only 2, which might suggest that the words are very similar, but in reality, they have no meaningful connection. The Levenshtein distance between "cat" and "dog" is 3 because each letter needs to be substituted to transform one word into the other. This highlights the limitation of using ASCII sums, which can lead to incorrect conclusions about word similarity.

➔ It Fails to Handle Word Length Differences

Using the summation of ASCII codes does not handle differences in word length effectively. Longer words naturally have larger ASCII sums, even if they are otherwise similar to shorter words. For example, the ASCII sum for "book" is 427, while for "bo" it is 209. The difference is 218, which is large, but the words "book" and "bo" are clearly related. The Levenshtein distance between them is only 2 (representing two deletions), which more accurately reflects their similarity.

➔ Levenshtein Distance Captures Real Spelling Errors

Levenshtein distance is specifically designed to capture common types of spelling errors, such as substitutions, insertions, and deletions, which are the most frequent errors in real-world text. It accurately reflects the number of edits needed to change one word into another. By contrast, summing ASCII values does not take into account these common spelling changes and offers no insight into the structure of the words or the types of errors involved.

Conclusion

While summing ASCII values may seem like a simple way to measure word similarity, it is not a suitable replacement for Levenshtein distance. The sum of ASCII values ignores the position of characters, can lead to misleading results when different words have similar sums, and does not handle word length differences well. Most importantly, it does not reflect the actual types of errors we encounter in real spelling mistakes, such as substitutions, insertions, and deletions. Levenshtein distance is much better suited for tasks like spell checking because it captures the structural changes between words and provides a meaningful measure of the effort required to transform one word into another.