

Self-reflection (Week 2)

Through my testing with different numbers for changing the “x” value, I noticed some clear patterns in the results. When I changed x to a smaller number, such as 5, the output often showed simple and recognizable words as the last word of each line, including single-letter words like “a” or “l.” As I continued increasing the value of x, there was a higher chance that the final word of a line would turn into a symbol or a strange single character, such as “-” or letters like “w” and “t.”

However, no matter whether x was small or large, the poem did not become more logical overall. The meaning of the text still felt unclear, and many of the lines did not connect with each other in a meaningful way. In most cases, the lines created through the P+n method resulted in nonsense. This made it clear that adjusting x mainly affects how readable or broken the final word looks, rather than improving the poem’s overall meaning.

For the P+7 technique where all nouns are replaced with their seventh-highest probability alternatives, I think the code should start by using or creating a noun database through noun word tagging. Then, the poem would be tokenized by words, using spaces and line breaks. The last word of each line would then be selected, and it would be checked against the noun database to determine whether it is a noun. Once a noun is found, the language model would generate a ranked list of probable replacement words, with the list consisting only of nouns from the database. After that, the seventh word on the list would be selected. This replacement would first save the original noun’s position in the line as an array or a range, and when it comes to the time to paste the new word, the selected noun would be inserted in the exact position as the original. This ensures that the noun changes align with the P+7 technique while maintaining the original poem’s structure.