Assignment-04 Text Completion

Description:

The 1980s saw a shift from Natural Language Processing techniques aiming to codify the grammatical rules of natural language towards techniques aiming to use statistical models to generate text. One early idea which technically isn't "AI" seeing as it is "memorizing" the training data and yet introduces us to the power contained in statistical techniques of text generation is the idea of Markov chains. Write a python function `generate(filename: str, start_words: list[str], chain_length: int, num_generated: int) -> str` which takes a filename, a chain length, a list of start words which has to be exactly as long as the `chain_length` (why?), and an integer `num_generated` and returns a sentence `num_generated` words long which sounds similar to the text contained in `filename`.

Approaching GPT

1. Understanding Markov Chains.
2. Approaching the concept of Text Prediction / Text Generation with Markov Chain Concepts.
3. Steps to build a computer program that essentially predicts the next words.
4. Steps:
    a. Building a text corpus and cleaning.
    b. Writing intermediary functions implementing the markov chain rule/concept, essentially creating a dictionary of inter-related words.
    c. Based on provided starting words, chain_length and required number words to be further generated, a string is resulted as an output.
5. Sample Test Cases given by GPT are run and tested in colab.
6. My own sample test cases are also given to the code and tested.
7. Asked to handle errors and exceptions.
8. Asked GPT to encapsulate all the functions, corpus and sample test cases into a class for concise approach.
9. Documented the conversation in the repository.