Homework Assignment No. 3

Text Processing and OCR in Practice

Corpus Selection and Research Question

I am interested in investigating how literary representations of attention and distraction have evolved throughout the twentieth century. Starting with Hermann Hesse's Siddhartha as an anchor point, one could potentially examine a curated corpus spanning from early modernist works through contemporary digital-age fiction, analyzing how the language around attention, concentration, and mental fragmentation has shifted alongside technological and social changes. This approach treats literature as a linguistic archive of our evolving relationship with attention and distraction.

The methodology could center around building a comprehensive lexicon of terms related to attention and distraction, then tracking their frequency and usage patterns across different decades. I am particularly interested in identifying which traditional words around focus and contemplation decline over time, while new vocabulary emerges - terms like "multitasking," "overwhelm," or "screen time" that reflect contemporary concerns. Simple frequency analysis can reveal these temporal shifts, showing how literary language adapts to new forms of distraction. Following this lexical analysis, I could use topic modeling to identify broader thematic patterns, examining how clusters of attention-related concepts evolve across different literary periods and technological contexts throughout the 20th century. This combination of straightforward counting techniques and topic modeling should provide a clear picture of how our literary discourse around attention has fundamentally changed.

Sampling Techniques Comparison

For the sampling analysis, I implemented both random and systematic sampling methods to extract chapters from the Siddhartha corpus. The random sampling approach used Python's `random.sample()` to select 10 chapters without replacement, essentially shuffling all available chapters and taking the first 10 from the shuffled list. This method gives every chapter an equal probability of selection but doesn't guarantee representation across different parts of the text.

The systematic sampling method was slightly more involved. I calculated a step size by dividing the total number of chapters by the desired sample size. For systematic sampling, I then randomly selected a starting point: if the step size was 1, I could start from any position in the chapter list; otherwise, I selected a starting point within the first step interval.. From there, I selected every nth chapter based on the calculated step size:

```
step_size = max(1, len(chapters) // sample_size)
start_index = random.randint(0, step_size - 1)
```

This approach ensures a more uniform distribution across the text, but maintains the random element through the selection of the starting point. I definitely could have done more sophisticated stratification by chapter length or position, but given the relatively small corpus size and the need to keep the implementation straightforward, I stuck with these two standard approaches. The comparison between the methods showed some interesting differences. In running the sampling cell, there was never a huge disparity in word count, generally a difference of around 300$\sim$ words, but the difference looked greater in tokens and the largest in char count.

OCR Implementation and Results

This process worked without fail. However, my approach was quite naive. I simply defined a rectangle for the contents I wanted to capture using margins. Although this is not a scalable solution, it did the job in this context. (I was spending too much time coding.) The content was preserved but reads from top to bottom across the entire document as seen here, as opposed to columns on the page.