Experiment Table

Semantic Chunking Test Model	Evaluation Methods:
Chonkie	**CosineSimilarity**
SentenceTranseformer	**NLI**
LLM	**Rouge Score**
LLM_Customized	**LLM**

Conclusion:

Using libraries like Chonkie or similar tools to fully customize the entire process requires relatively fewer resources during the execution phase but demands a significant amount during the preparation phase. Additionally, adjustments are always necessary depending on the structure and nature of the text. Nevertheless, the results were excellent.

Methods like SentenceTransformer easily produce decent results and offer models specialized for certain domains. However, the results lacked sophistication.

There are two primary approaches to using LLMs: one that relies solely on the basic capabilities of the LLM for chunking, and another that customizes the LLM chunking process using prompts and additional logic.

The first approach, relying solely on LLM capabilities, is relatively simple to implement and often yields a larger amount of textual output. However, the results exhibited significant variability.

For customized LLMs, the process required significant time for coding and debugging. In this approach, I employed a specific method that incrementally merged chunks with similar contexts. However, during this process, some chunks became disproportionately large, causing errors as the LLM could not handle them.

To resolve this issue, I tried various methods, such as limiting chunk size, restricting the number of chunks, and batch processing. After converting the vector store to Excel for verification, I found that some approaches were structured in a way that was unsuitable for the LLM to generate responses. Ultimately, I adopted a method that splits chunks into smaller pieces once they reach a certain size. This allowed me to overcome the token-size limitations of the LLM without losing text.

Using the open-source LLM, Mistral AI, processing a 50-page PDF with a large amount of

text took over 4 hours due to the waiting time for requests. I also tested GPT-3.5, which took about 30 minutes, but using it requires additional costs for OpenAI's API.

Theoretically, I predict that the customized LLM approach I tested last would deliver the best performance. However, the quantitative and qualitative results of Chonkie were also satisfactory. That said, Chonkie's performance is likely to vary depending on the nature of the document and the style of the text, making it difficult to draw a definitive conclusion from this alone.

Ultimately, to derive a convincing conclusion, all four models must undergo evaluation using four quantitative methods as well as qualitative assessments conducted by observers. These tests need to be repeated sufficiently to record and observe results comprehensively. Additional time and resources will be necessary for this, and there remains substantial room for experimentation with hybrid structures that combine language embedding models, specific libraries, and LLMs. These approaches would be applicable not only to model development but also to the development of evaluation methods.

Lastly, it's important to be cautious about over-relying on the four quantitative evaluation methods mentioned earlier. Until more definitive methods are identified, they should serve only as references. Final evaluations must always be conducted by humans.

To further refine this model and evaluation methodology, a deeper exploration of how libraries, embedding models, and LLMs generate results is essential.