Dealing long input issue

1. **Document Chunking:**

Split the long document into smaller chunks or sections, and process each chunk independently.

After generating summaries for individual chunks, then combine these summaries to create a summary for the entire document.

1. **Extraction and Abstraction:**

Use an extractive summarization model for the initial processing of the document. Extractive summarization methods select important sentences or phrases from the document without rewriting them.

Then pass this condensed version to the abstractive summarization model for further summarization.

1. **Summary Compression:**

If the summaries are still too long, consider applying additional compression techniques. This might involve further shortening sentences or using advanced compression algorithms to condense the information.

1. **Key Sentence Identification:**

Identify key sentences or paragraphs in the document that are likely to contain important information. Focus on summarizing these key sections.

Use techniques like keyword extraction, named entity recognition, or even simple heuristics to identify important content.

1. Pre-processing:

Before feeding the document into the model, pre-process it by removing unnecessary information, such as boilerplate text, footnotes, or irrelevant sections. This will reduce the overall length of the document.

Blogs referred:

1. <https://medium.com/@eboraks/summarize-and-knowledge-extraction-of-large-documents-using-lsa-and-llm-2dbee4142da4>
2. Using LangChain to divide the text into chunks, summarize them separately, stitch them together, and re-summarize to get a consistent answer
3. Use statistical methods like Latent Semantic Analytics (LSA) to remove excess (redundant) information from the text. LSA is good at distilling the essential information, but its output is difficult for human to read. But, LLM can take the LSA output, make sense of it, and write an excellent summary.

(Recommended for input tokens less than 20k)