# LLMs for SLR Data Extraction

* **Read in pdf files:** PDFs cannot be read directly so we need to use a specialised tool. Need to investigate which tool can provide the best results. We would need to transform the PDF into a format we can read into our application (Typically raw text).
* **Chunking Documents:** The documents need to be divided into discrete units. There are different approaches and it is worth testing different ones. (Langchain is a good source)
* **Choose an appropriate LLM:** Many choices here, from general purpose massive models like those by OpenAI or smaller purpose built models provided by Ollama that may be run locally. Local models add the benefit of preserving privacy (i.e. keeping sensitive documents in your organisation) potentially at the expense of quality.
* **Produce Embeddings for document chunks:** Text chunks are processed by the selected LLM to produce a vector representation of the information within as ‘understood’ by the model.
* **Storage:** The documents embeddings are stored in a system capable of performing fast vector operations to facilitate later retrieval functionality.
* **Querying by field:** Each of the baseline characteristics and clinical outcomes specified by the customer are processed by the same LLM to produce compatible embeddings. These embeddings are then used to retrieve the closest matching top N chunks from each document by means of their associated embeddings.
* **Extraction and/or synthesis step:** Selecting the data to be associated to the query field
  + **Top-1 chunk:** The chunk that most closely relates to the query can be selected for the field. Avoids hallucination issues, but the usefulness/representativeness of the text depends on the chunking approach.
  + **Augmentation:** Using an LLM prompt a top-n set of results can be transformed into an appropriate value for the field. However there is a danger for hallucinations.
* **Formatting and output:** The results are to be compiled in a tabular format (CSV or Excel) with appropriate headings

**Notes:**

* **Choice of PDF processing tool** can lead to very different results. Tables may need to be dealt with a specific tool.
* **Choice of LLMs** can impact the quality of the matching and Augmentation. Potentially two different ones could do a better job. One trained on medical corpora to produce embeddings used for the matching stage. Another one more general purpose capable of synthesising an appropriate answer from top matches.
* **Extraction using JSON schemas.** Another option for extraction, is to provide a JSON schema as part of a prompt that defines some rigid format to follow. This offloads the whole process of extraction to the LLM. However it is to be seen how good would be the output, and what is the impact of hallucinations.
* **Evaluation.** For a problem with so many moving parts, it would be beneficial to perform an evaluation that gives us an statistical indication of quality. This is particularly important for a production-ready product. An appropriate evaluation could take into account a human made gold-standard to evaluate against, such as a previously completed SLR. Along with evaluation metrics. (If using the OpenAI platform, it has an interesting evaluation option to try out).
* **Structured input data?:** I would explore the use of structured XML style output from the PDF reader, and whether it could be used to give cues to any of the stages. For example, XML structures could be leveraged, rather than a set of text following a sliding window, for a chunking strategy. It could help with dealing with table structures too, helping to associate cell values to relevant headers.