**Implementation:**

1. **Data Extraction:**

The code reads text files (specifically Microsoft Word .docx documents) and breaks them into individual paragraphs. This is essential because the later clustering techniques work best with smaller, self-contained units of meaning. Information extraction and analysis are much easier to perform when dealing with well-defined segments of text instead of one big block.

1. **Data pre-processing**

Pre-processing usually involves removing extra character, misspellings or converting text to lowercase. In our case, we remove the punctuations and ensure that the paragraphs are split uniformly. We also remove misspellings and unrecognized text (Languages other than English) from the extracted paragraphs. Cleaning up text makes it more uniform. This helps the clustering to be more precise.

1. **Word-Embedding**

The code uses a powerful model called the Universal Sentence Encoder. It transforms each paragraph into a set of numbers (a vector). Vectors that represent similar paragraphs will be positioned closer together in a "conceptual space." This is very helpful in this case as this research focuses on unstructured data that usually does not follow a pattern. LLM’s are trained on many different types of documents and text making it a better choice compared to NLP for this research.

1. **K means Clustering**

K-means is a classic algorithm for finding groups when performing unsupervised learning. Clustering helps discover patterns in the paragraphs. Paragraphs in the same cluster probably address similar topics or themes.

This algorithm is the perfect fit for this research as we are not clear about the patterns or the topics in the documents. The algorithm takes the paragraph vectors as input and tries to find "centers" where similar paragraphs are clustered. Each paragraph gets assigned to its closest center. The number of clusters is decided beforehand.

1. **Hierarchical Clustering**

At this stage the algorithm takes the regular K-means clustering and goes deeper. It takes one cluster at a time and applies K-means again within that cluster. This creates layers of groups—big groups broken down into smaller ones.

1. **Rule based Relevant Information Extraction**

Here we have used pattern-matching rules to find things like PAN and Aadhar card numbers within the paragraphs. It also has a way to spot phrases like "VENDOR" or "PURCHASER" to tag paragraphs. This is where we finally get the information we are looking for. The clustering helps to focus on the correct parts of the document, and then these rules extract the final details.

**Testing:**

1. **Training KNN model based on Kmeans outputs**

Since we have used a dataset that is both private and not an established dataset, there are no standard testing techniques that can be applied on the results. So, in this research we have tested the clustering by training a KNN model based on the paragraph embeddings and the obtained Kmeans labels. The KNN model classifies the embeddings based on the differences in the Kmeans labels.

1. **Testing obtained KNN results:**

After the KNN model is trained, other documents are given as input and the predicted Kmeans labels are obtained as results. Here, the new KNN results are predicted Kmeans labels based on the previous training. The results are stored as a csv file that contains both the initial Kmeans labels and predicted Kmeans labels.

1. **Accuracy measure**

The results obtained are compared with the actual Kmeans labelling and an approximate accuracy score is calculated manually.