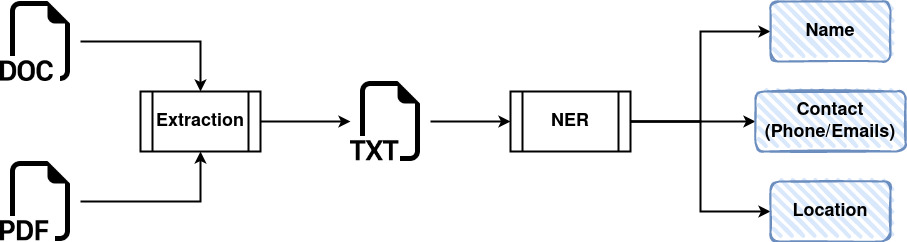
**Report**

Problem Statement:

Given a text document of formats including text, docx, or pdf the objective is to extract key entities from it. The entities to be recognized are name, location, email address and phone number. We need not find it in context to any person mentioned but all such mentioned entities in a list. The document will be input to the application and the entities, list of each of the required ones, are the expected output.



Approach:

As shown in the diagram above, this task entails two major items:

1. Validating extracting, and processing the input document
2. Using appropriate pre-trained NLP based NER model, for extracting the entities, post-processing the output and saving it in appropriate file for later use.

Process:  
For the first task, we take the input from the user through the command line and validate it. Only the acceptable input (one of the file types – TXT, DOCX, PDF). For this assignment, I’ve considered the PDF files which do not require OCR/ICR capabilities.

Once validated and ascertained that the file has text data, we extract it with the help of libraries for the respective file types (PDF – PyMuPDF, DOCX – Python-docx).

As a result, we get raw data extracted from the document which is than passed a cleaning function which removes all the junk characters, strips it and returns the clean text as a output.

The extracted text, as a sequence of utf-8 characters, is passed on the NER model. For this task, I chose the popular BERT pre-trained NER model (bert-base-NER). We use a PyTorch based pipeline for leveraging BERT from the transformers library. We create a pipeline by invoking a tokenizer using BERT base NER for AutoModel for NER Token Classification. Since it is a pretained model trained on the CoNLL dataset, it can identify 4 entities (including Name and location present in the text). The result we get back is the raw inference of the model which comes as a list of words (in tokens and its corresponding classification of tags). The tags use a very specialized nomenclature (eg. I-PER, B-PER, I-LOC, B-LOC) fpr the two entities person and location. `B-` marks the start of the recognized entity and `I-` marks as the end of the recognized entity. This is the result reutrned by the model

The next step is to clean the result or to post-process it. We combine the entities with their respective `B-` and `I-` tags and return it as a single entity back as the recognized person or location. As a result we have a pruned list of individual entities in the list of the persons and location.

For the entities such as email and phone numbers, I use REGEX. For the phone number regex we simply check if the sequence of numbers are not more than 10 digits and in the form (xxx xxx xxxx) , (xxx-xxx-xxxx), (xxxxxxxxxx). They are separated by alternation. For the email part, it is checking for a few characters (one or more) before `@` a few after (one or more) followed by a `.` and a few at the end (one or more).

After all the entities are recognized they are collated together and returned from the class which is than printed out in a json file and logged in the log file.

About the Model:

- **bert-base-NER (Transformer based)**

Use: Fine Tuned BERT model that is ready to use for NER. Trained for four entities (PER, ORG, LOC, MISC). It is a cased model that is trained on CoNLL-2003 Named Entity Dataset

NER Capabilities: Limited by training dataset and thus not generalizable. Performs badly on the custom input sentences. The model is highly biased towards names and location from the western people. Transformers based models take a lot of time for producing inferences.