**Introduction:**

We are assigned with the task of designing an AI chatbot to make responses from the provided data.

**Dataset:**

In this document we guys are here to discuss the loading and preprocessing of the dataset to make an effective chatbot.

**Source of Data:**

We made our own dataset for this chatbot . Thus we can customise the chatbot by customising the dataset.

**Data Processing:**

In this process we have made a pdf of words and then we turn them into a token so that it can be converted into vectors . thus it is the key point involved in this process .

**Python Program for data initialization and processing:**

!nvidia-smi

!pip install -Uqqq pip --progress-bar off

!pip install torch==2.0.1 --progress-bar off

!pip install transformers==4.31.0 --progress-bar off

!pip install langchain==0.0.266 --progress-bar off

!pip install chromadb==0.4.5 --progress-bar off

!pip install pypdf==3.15.0 --progress-bar off

!pip install xformers==0.0.20 --progress-bar off

!pip install sentence\_transformers==2.2.2 --progress-bar off

!pip install InstructorEmbedding==1.0.1 --progress-bar off

!pip install pdf2image==1.16.3 --progress-bar off

!wget -q https://github.com/PanQiWei/AutoGPTQ/releases/download/v0.4.1/auto\_gptq-0.4.1+cu118-cp310-cp310-linux\_x86\_64.whl

!pip install -qqq auto\_gptq-0.4.1+cu118-cp310-cp310-linux\_x86\_64.whl --progress-bar off

!sudo apt-get install poppler-utils

!mkdir pdfs

from pdf2image import convert\_from\_path

from transformers import AutoTokenizer, TextStreamer, pipeline

!gdown 1v-Rn1FVU1pLTAQEgm0N9oB6cExMoebZr -O pdfs/tesla-earnings-report.pdf

!gdown 1Xc890jrQvCExAkryVWAttsv1DBLdVefN -O pdfs/nvidia-earnings-report.pdf

!gdown 1Epz-SQ3idPpoz75GlTzzomag8gplzLv8 -O pdfs/meta-earnings-report.pdf

meta\_images = convert\_from\_path("/content/pdfs/dataf7b.pdf", dpi=88)

meta\_images[0]

!rm -rf "db"

loader = PyPDFDirectoryLoader("pdfs")

docs = loader.load()

len(docs)

embeddings = HuggingFaceInstructEmbeddings(

model\_name="hkunlp/instructor-large", model\_kwargs={"device": DEVICE}

)

text\_splitter = RecursiveCharacterTextSplitter(chunk\_size=1024, chunk\_overlap=64)

texts = text\_splitter.split\_documents(docs)

len(texts)

%%time

db = Chroma.from\_documents(texts, embeddings, persist\_directory="db")

**Conclusion:**

Thus this is the process involved in the process of data initialization and processing in the dataset of the process.