


```
import warnings
warnings.filterwarnings("ignore")
```

```
!pip install --quiet langchain_openai transformers sentence_transformers torch pipeline langchain_community
!pip install --quiet langchain unstructured[all-docs] pydantic lxml langchainhub fastapi
!pip install pillow==8.3.2
!pip install typer==0.3.0
```


 [Show hidden output](#)

```
!sudo apt-get install poppler-utils tesseract-ocr
```

 Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
tesseract-ocr is already the newest version (4.1.1-2.1build1).  
poppler-utils is already the newest version (22.02.0-2ubuntu0.4).  
0 upgraded, 0 newly installed, 0 to remove and 45 not upgraded.

```
import openai
import langchain
from langchain.document_loaders import PyPDFLoader
from langchain.text_splitter import CharacterTextSplitter
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain_community.embeddings import HuggingFaceEmbeddings
from langchain.vectorstores import Pinecone
from langchain.llms import OpenAI
import os
```


```
from unstructured.partition.pdf import partition_pdf
elements = partition_pdf("/content/manual.pdf",
                        extract_images_in_pdf=True,
                        infer_table_structure=True,
                        chunking_strategy="by_title",
                        max_characters=1000,
                        new_after_n_chars=1000,
                        strategy="hi_res"
                        )
```

 Some weights of the model checkpoint at microsoft/table-transformer-structure-recognition were not used when initializing TableTransformerForObjectDetection from the checkpoint of a model trained on another task or  
- This IS expected if you are initializing TableTransformerForObjectDetection from the checkpoint of a model trained on another task or  
- This IS NOT expected if you are initializing TableTransformerForObjectDetection from the checkpoint of a model that you expect to be

```
# Create a dictionary to store counts of each type
category_counts = {}
```

```
for element in elements:
    category = str(type(element))
    if category in category_counts:
        category_counts[category] += 1
    else:
        category_counts[category] = 1
```

```
# Unique_categories will have unique elements
unique_categories = set(category_counts.keys())
category_counts
```

 {"<class 'unstructured.documents.elements.CompositeElement'>": 36,  
 "<class 'unstructured.documents.elements.Table'>": 6,  
 "<class 'unstructured.documents.elements.TableChunk'>": 26}

```

from lxml import html
from pydantic import BaseModel
from typing import Any, Optional
class Element(BaseModel):
    type: str
    text: Any

# Categorize by type
categorized_elements = []
for element in elements:
    if "unstructured.documents.elements.Table" in str(type(element)):
        categorized_elements.append(Element(type="table", text=str(element)))
    elif "unstructured.documents.elements.CompositeElement" in str(type(element)):
        categorized_elements.append(Element(type="text", text=str(element)))

```

```

# Tables
table_elements = [e for e in categorized_elements if e.type == "table"]
print(len(table_elements))
print(table_elements[0].text)
# Text
text_elements = [e for e in categorized_elements if e.type == "text"]
print(len(text_elements))

```

```

↩ 32
Maintenance Record Sheet
36

```

```

model_name="sentence-transformers/all-mpnet-base-v2"
embeddings=HuggingFaceEmbeddings(model_name=model_name)

```

```

vectors=embeddings.embed_query("How are you?")
len(vectors)

```

```

↩ 768

```

```

!pip install langchain-chroma

```

```

↩

```

```

Requirement already satisfied: watchfiles>=0.13 in /usr/local/lib/python3.10/dist-packages (from uvicorn[standard]>=0.18.3->chromadb)
Requirement already satisfied: websockets>=10.4 in /usr/local/lib/python3.10/dist-packages (from uvicorn[standard]>=0.18.3->chromadb)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio->httpx>=0.23.0->fastapi>1, >=0.95
Requirement already satisfied: cachetools<6.0, >=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth>=1.0.1->kubernetes
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth>=1.0.1->kubernetes)
Requirement already satisfied: rsa<5, >=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth>=1.0.1->kubernetes>=28.1.0-
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0, >=0.16.4->tokenizers>=0.
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0, >=0.16.4->tokeni
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata<=7.0, >=6.0->opentele metr
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.11.0->typer>=0.9.0->ch
Requirement already satisfied: pygments<3.0.0, >=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.11.0->typer>=0.9.0->
Requirement already satisfied: humanfriendly>=9.1 in /usr/local/lib/python3.10/dist-packages (from coloredlogs->onnxruntime>=1.14.1->
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->onnxruntime>=1.14.1->chromadb<0.6
Requirement already satisfied: mdurl~0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->type
Requirement already satisfied: pyasn1<0.7.0, >=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-au

```

```

!pip install chromadb
import chromadb
chroma_client = chromadb.Client()
#collection = chroma_client.create_collection(name="summaries")

```

```

Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from kubernetes>=28.1.0->chromadb) (1.16.0)
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.10/dist-packages (from kubernetes>=28.1.0->chromadb)
Requirement already satisfied: google-auth>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from kubernetes>=28.1.0->chromadb) (2.2
Requirement already satisfied: websocket-client!=0.40.0, !=0.41.*, !=0.42.*, >=0.32.0 in /usr/local/lib/python3.10/dist-packages (from k
Requirement already satisfied: requests-oauthlib in /usr/local/lib/python3.10/dist-packages (from kubernetes>=28.1.0->chromadb) (1.3.
Requirement already satisfied: oauthlib>=3.2.2 in /usr/local/lib/python3.10/dist-packages (from kubernetes>=28.1.0->chromadb) (3.2.2)
Requirement already satisfied: urllib3>=1.24.2 in /usr/local/lib/python3.10/dist-packages (from kubernetes>=28.1.0->chromadb) (2.0.7)
Requirement already satisfied: coloredlogs in /usr/local/lib/python3.10/dist-packages (from onnxruntime>=1.14.1->chromadb) (15.0.1)
Requirement already satisfied: flatbuffers in /usr/local/lib/python3.10/dist-packages (from onnxruntime>=1.14.1->chromadb) (24.3.25)
Requirement already satisfied: protobuf in /usr/local/lib/python3.10/dist-packages (from onnxruntime>=1.14.1->chromadb) (3.20.3)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from onnxruntime>=1.14.1->chromadb) (1.12)
Requirement already satisfied: deprecated>=1.2.6 in /usr/local/lib/python3.10/dist-packages (from opentelemetry-api>=1.2.0->chromadb)
Requirement already satisfied: importlib-metadata<=7.0, >=6.0 in /usr/local/lib/python3.10/dist-packages (from opentelemetry-api>=1.2.
Requirement already satisfied: googleapis-common-protos==1.52 in /usr/local/lib/python3.10/dist-packages (from opentelemetry-exporter
Requirement already satisfied: opentelemetry-exporter-otlp-proto-common==1.24.0 in /usr/local/lib/python3.10/dist-packages (from open
Requirement already satisfied: opentelemetry-proto==1.24.0 in /usr/local/lib/python3.10/dist-packages (from opentelemetry-exporter-ot
Requirement already satisfied: opentelemetry-instrumentation-asgi==0.45b0 in /usr/local/lib/python3.10/dist-packages (from openteleme
Requirement already satisfied: opentelemetry-instrumentation==0.45b0 in /usr/local/lib/python3.10/dist-packages (from opentelemetry-i
Requirement already satisfied: opentelemetry-semantic-conventions==0.45b0 in /usr/local/lib/python3.10/dist-packages (from openteleme
Requirement already satisfied: opentelemetry-util-http==0.45b0 in /usr/local/lib/python3.10/dist-packages (from opentelemetry-instrum
Requirement already satisfied: setuptools>=16.0 in /usr/local/lib/python3.10/dist-packages (from opentelemetry-instrumentation==0.45b
Requirement already satisfied: wrapt<2.0.0, >=1.0.0 in /usr/local/lib/python3.10/dist-packages (from opentelemetry-instrumentation==0.
Requirement already satisfied: asgiref<=3.0 in /usr/local/lib/python3.10/dist-packages (from opentelemetry-instrumentation-asgi==0.45
Requirement already satisfied: monotonic>=1.5 in /usr/local/lib/python3.10/dist-packages (from posthog>=2.4.0->chromadb) (1.6)
Requirement already satisfied: backoff>=1.10.0 in /usr/local/lib/python3.10/dist-packages (from posthog>=2.4.0->chromadb) (2.2.1)
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic>=1.9->chromadb) (0.7.
Requirement already satisfied: pydantic-core==2.18.2 in /usr/local/lib/python3.10/dist-packages (from pydantic>=1.9->chromadb) (2.18.
Requirement already satisfied: charset-normalizer<4, >=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.28->chromadb) (3
Requirement already satisfied: idna<4, >=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.28->chromadb) (3.7)
Requirement already satisfied: huggingface-hub<1.0, >=0.16.4 in /usr/local/lib/python3.10/dist-packages (from tokenizers>=0.13.2->chro
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.10/dist-packages (from typer>=0.9.0->chromadb) (8.1.7)
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from typer>=0.9.0->chromadb) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.10/dist-packages (from typer>=0.9.0->chromadb) (13.7.1)
Requirement already satisfied: h11>=0.8 in /usr/local/lib/python3.10/dist-packages (from uvicorn[standard]>=0.18.3->chromadb) (0.14.0)
Requirement already satisfied: httptools>=0.5.0 in /usr/local/lib/python3.10/dist-packages (from uvicorn[standard]>=0.18.3->chromadb)
Requirement already satisfied: python-dotenv>=0.13 in /usr/local/lib/python3.10/dist-packages (from uvicorn[standard]>=0.18.3->chroma
Requirement already satisfied: uvloop!=0.15.0, !=0.15.1, >=0.14.0 in /usr/local/lib/python3.10/dist-packages (from uvicorn[standard]>=0
Requirement already satisfied: watchfiles>=0.13 in /usr/local/lib/python3.10/dist-packages (from uvicorn[standard]>=0.18.3->chromadb)
Requirement already satisfied: websockets>=10.4 in /usr/local/lib/python3.10/dist-packages (from uvicorn[standard]>=0.18.3->chromadb)
Requirement already satisfied: dnspython>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from email_validator>=2.0.0->fastapi>=0.9
Requirement already satisfied: cachetools<6.0, >=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth>=1.0.1->kubernetes
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth>=1.0.1->kubernetes)
Requirement already satisfied: rsa<5, >=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth>=1.0.1->kubernetes>=28.1.0-
Requirement already satisfied: anyio in /usr/local/lib/python3.10/dist-packages (from httpx>=0.23.0->fastapi>=0.95.2->chromadb) (3.7.
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.10/dist-packages (from httpx>=0.23.0->fastapi>=0.95.2->chromad
Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from httpx>=0.23.0->fastapi>=0.95.2->chromadb) (1.
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0, >=0.16.4->tokenizers>=0.
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0, >=0.16.4->tokeni
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata<=7.0, >=6.0->opentele metr
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2>=2.11.2->fastapi>=0.95.2->chro
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.11.0->typer>=0.9.0->ch
Requirement already satisfied: pygments<3.0.0, >=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.11.0->typer>=0.9.0->
Requirement already satisfied: humanfriendly>=9.1 in /usr/local/lib/python3.10/dist-packages (from coloredlogs->onnxruntime>=1.14.1->
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->onnxruntime>=1.14.1->chromadb) (1
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio->httpx>=0.23.0->fastapi>=0.95.2-
Requirement already satisfied: mdurl~0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->type
Requirement already satisfied: pyasn1<0.7.0, >=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-au

```

```

from transformers import pipeline
import torch
from langchain.llms import HuggingFacePipeline
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM
LLM_Model="google/flan-t5-large"
tokenizer=AutoTokenizer.from_pretrained(LLM_Model)
model=AutoModelForSeq2SeqLM.from_pretrained(LLM_Model, torch_dtype=torch.float32)
pipe=pipeline(
    "text2text-generation",
    model=model,
    tokenizer=tokenizer,
    max_length=300,
    temperature=0,
    top_p=0.95,
    repetition_penalty=1.15
)
llm=HuggingFacePipeline(pipeline=pipe)
llm

```

```

/usr/local/lib/python3.10/dist-packages/langchain_core/_api/deprecation.py:119: LangChainDeprecationWarning: The class `HuggingFacePipeline` is deprecated.
HuggingFacePipeline(pipeline=<transformers.pipelines.text2text_generation.Text2TextGenerationPipeline object at 0x7e0227f07340>)

```

```
llm("What is the capital of India")
```

```

/usr/local/lib/python3.10/dist-packages/langchain_core/_api/deprecation.py:119: LangChainDeprecationWarning: The class `HuggingFacePipeline` is deprecated.
'calcutta'

```

```

from langchain_core.prompts import ChatPromptTemplate
from langchain_core.output_parsers import StrOutputParser
prompt_text="""You are an assistant tasked with summarizing tables and text provided in the Car Manual. \
Give a concise summary of the table or text.Table or Text chunk: {element}"""
prompt=ChatPromptTemplate.from_template(prompt_text)
summarize_chain={ "element": lambda x: x } | prompt | llm | StrOutputParser(max_length=500)

```

```

!pip install langchain_core
from langchain_core.prompts import ChatPromptTemplate
prompt2=ChatPromptTemplate.from_template("""
You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include
If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know,
If the answer is from the tables, Display it in table format\
Provide answers in the human readable format.
<context>
{context}
</context>
Question: {input}
""")

```

```

Requirement already satisfied: langchain_core in /usr/local/lib/python3.10/dist-packages (0.2.2)
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.10/dist-packages (from langchain_core) (6.0.1)
Requirement already satisfied: jsonpatch<2.0,>=1.33 in /usr/local/lib/python3.10/dist-packages (from langchain_core) (1.33)
Requirement already satisfied: langsmith<0.2.0,>=0.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain_core) (0.1.63)
Requirement already satisfied: packaging<24.0,>=23.2 in /usr/local/lib/python3.10/dist-packages (from langchain_core) (23.2)
Requirement already satisfied: pydantic<3,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain_core) (2.7.1)
Requirement already satisfied: tenacity<9.0.0,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain_core) (8.3.0)
Requirement already satisfied: jsonpointer>=1.9 in /usr/local/lib/python3.10/dist-packages (from jsonpatch<2.0,>=1.33->langchain_core) (2.3)
Requirement already satisfied: orjson<4.0.0,>=3.9.14 in /usr/local/lib/python3.10/dist-packages (from langsmith<0.2.0,>=0.1.0->langchain_core) (3.9.14)
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.10/dist-packages (from langsmith<0.2.0,>=0.1.0->langchain_core) (2.31.0)
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain_core) (0.6.0)
Requirement already satisfied: pydantic-core==2.18.2 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain_core) (2.18.2)
Requirement already satisfied: typing-extensions>=4.6.1 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain_core) (4.6.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langsmith<0.2.0,>=0.1.0->langchain_core) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langsmith<0.2.0,>=0.1.0->langchain_core) (3.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langsmith<0.2.0,>=0.1.0->langchain_core) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langsmith<0.2.0,>=0.1.0->langchain_core) (2024.2.2)

```

```

import uuid
from langchain.vectorstores import Chroma
from langchain.storage import InMemoryStore
from langchain.schema.document import Document
from langchain.embeddings import OpenAIEmbeddings
from langchain.retrievers.multi_vector import MultiVectorRetriever
from langchain_community.embeddings import HuggingFaceEmbeddings
vectorstore = Chroma(
    collection_name="summaries",
    embedding_function=embeddings
)

# The storage layer for the parent documents
store = InMemoryStore()
id_key = "doc_id"

# The retriever (empty to start)
retriever = MultiVectorRetriever(
    vectorstore=vectorstore,
    docstore=store,
    id_key=id_key,
)
retriever

↗ MultiVectorRetriever(vectorstore=<langchain_community.vectorstores.chroma.Chroma object at 0x7e02201231c0>, docstore=
<langchain_core.stores.InMemoryBaseStore object at 0x7e0220122da0>)

from langchain.schema.runnable import RunnablePassthrough
from langchain_core.output_parsers import StrOutputParser
tables = [i.text for i in table_elements]
print(tables)
table_summaries = summarize_chain.batch(tables, {"max_concurrency": 5})
table_summaries

↗ ['Maintenance Record Sheet', "'uBIS 'yoo' 'uBIS 'APY 48S dajeap Buildin -J8S JO OWEN auop sedoy jo sile}9q Arobaye9 djedoy swy 'ON OU',
['The Maintenance Record Sheet is a record of the maintenance of the vehicle.',
'“uBIS “APY 48S dajeap Buildin -J8S JO OWEN auop sedoy jo sile9q Arobaye9 djedoy swy “ON OU',
'The following are the names of the four cylinders of the car:',
'Exterior overview (i) 2-2 Exterior overview (ii) 2-3 Interior overview
..... 2-6 Instrument panel overview
..... 2-8 Instrument panel overview
..... 2-9 Engine compartment
.....2-10 Dimensions
.....2-11 Engine specification
.....2-11 Bulb wattage',
'The following information is provided in the Car Manual:',
'The following items are included in the following parts:',
'Windows
.....
'1. Antenna & doors 2. Fuel filler door 3. Rear ultrasonic sensors 4. High mounted stop light 5. Rear view camera 6. Rear window
wiper blade',
'The rear combination lamp is 9-81.',
'The hood is 5-37.',
'The following is the number of cylinders in the engine:',
'1. Antenna & doors 2. Doors 3. Fuel filler door 4. Rear ultrasonic sensors 5. High mounted stop light 6. Rear view camera 7.
Rear window wiper blade',
'The rear combination lamp is 9-81.',
'The door handle is located on the left side of the car.',
'The following parts are included in the manual:',
'The door handle is located on the left side of the car.',
'The following parts are included in the manual:',
'The following switches are used to control the following functions:',
'The manual has a lot of information about the car.',
'The warning light button is 8-2.',
'The following switches are used to control the following functions:',
'The manual has a lot of information about the car.',
'The warning light button is 8-2.',
'The following items are covered in this manual:',
'The radiator cap is 9-31.',
'The following items are included in the price of the car:',
'Item Displacement cc Bore x Stroke mm Firing order No. of cylinders 1.0 T-GDi 998 71 x 84 In-line 3 Petrol Engine Smartstream G1.2
1,197 71 x 75.6 In-line 4',
'Headlight (Low/High) Type A Turn signal light Position light (if equipped) Headlight (High/Low) Front Type B Turn signal light Day
running light & Position light (DRL&PSTN) Static bending lamp Daytime running light (LED) (if equipped) Side repeater lamp (if
equipped) Fog light (if equipped) Tail light (if equipped) Type A Tail/Stop light (if equipped) Turn signal light (if equipped) Back
up light (if equipped) Rear Tail light Type B Stop light Turn signal light Back up light High mounted stop light License plate light
Room lamp Interior Map light Luggage room light Light source H19 PY21W W5W LED PY21W LED H7 55W LED W5W HB4 W5W P21/5W PY21W W16W LED
LED PY21W W16W W5W W5W W5W FESTOON FESTOON FESTOON Wattage 55/60 21 5 LED 21 LED 55 LED 5 51 5 5/21 21 16 LED LED 21 16 5 5 8 8 10',
'The following table shows the mufflers and the hoods of the cars.',
'Item Full size tire 5 Door Tire size Wheel size 185/70R14 5.5J x 14 185/65R15 5.5J x 15 Inflation pressure kPa (psi) Normal load *1
Maximum load Front 230 (33) 230 (33) Rear 215 (31) 215 (31) Front 230 (33) 230 (33) Rear 230 (33) 230 (33) Wheel lug nut torque kgfm

```

```

(lbfft, Nm) 1113 (7994, 107127) 195/55R16 6.0J x 16 230 (33) 215 (31) 230 (33) 230 (33)',
'eer Engine oil? * ran and ref) |Petrol19 T-6Di 3.62 ACEA C2 SAE OW 20,APISN PLUS/SP Petrolsm artstream G12 3.42 or1SAC GE-6 ; APIGL-
4, SAE 70W M anual Petrol19 T-GDi 15-162 |_sk :HK SYN M TF 70W transn ssbn fui (ntelignentm anual -H K SHELL :SPIRAX S6 'isso
x1)|Petrolsm artsteam G12 1.3-1.4 2 GHME 70W M TF tans n fu) -GS CALTEX :GSMTFHD 70W Thtelignentm anual SAE J1704 DOT-4 LV, transn
'sion . 04925 CLASS-6, system actuator |Petol19 T-6b1 0.0822 |e vssiiis DOT-4, fui FM VSS116 DOT-3 APIGL-4,SAE 70W Dualclitch . _ HK D
DCTF GO -19 PLUS (Sk transm ssn fd [Pet0149 T-6D1 16-172 | cpRAX S6GHDE',
'The following table shows the mufflers and the hoods of the cars.'])

# Apply to texts
texts = [i.text for i in text_elements]
text_summaries = summarize_chain.batch(texts, {"max_concurrency": 5})

doc_ids = [str(uuid.uuid4()) for _ in texts]
summary_texts = [Document(page_content=s,metadata={id_key: doc_ids[i]}) for i, s in enumerate(text_summaries)]
retriever.vectorstore.add_documents(summary_texts)
retriever.docstore.mset(list(zip(doc_ids, texts)))

# Add tables
table_ids = [str(uuid.uuid4()) for _ in tables]
summary_tables = [Document(page_content=s,metadata={id_key: table_ids[i]}) for i, s in enumerate(table_summaries)]
retriever.vectorstore.add_documents(summary_tables)
retriever.docstore.mset(list(zip(table_ids, tables)))

from langchain.schema.runnable import RunnablePassthrough
from langchain_core.output_parsers import StrOutputParser
chain = (
    {"context": retriever, "input": RunnablePassthrough()}
    | prompt2
    | llm
    | StrOutputParser()
)
chain

{
  context: MultiVectorRetriever(vectorstore=<langchain_community.vectorstores.chroma.Chroma object at 0x7e02201231c0>, docstore=
<langchain_core.stores.InMemoryBaseStore object at 0x7e0220122da0>),
  input: RunnablePassthrough()
}
| ChatPromptTemplate(input_variables=['context', 'input'], messages=[HumanMessagePromptTemplate(prompt=PromptTemplate(input_variables=
['context', 'input'], template='\nYou are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while
being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please
ensure that your responses are socially unbiased and positive in nature.If a question does not make any sense, or is not factually
coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false
information.If the answer is from the tables, Display it in table formatProvide answers in the human readable
format.\n<context>\n{context}\n</context>\nQuestion: {input}\n'))]]
| HuggingFacePipeline(pipeline=<transformers.pipelines.text2text_generation.Text2TextGenerationPipeline object at 0x7e0227f07340>)
| StrOutputParser()

chain.invoke("What is Vehicle Identification Number(VIN)?")

Token indices sequence length is longer than the specified maximum sequence length for
'The vehicle identification number (VIN) is the number used in registering your car an
d in all legal matters pertaining to its ownership, etc. The number is punched on the
floor under the front right seat. To check the number, pull out the slit part of carpe
t.nn2-16nnVEH TLECERTFLATDN LABEL (IF EQUIPPED)nnS BS 224nnThe vehicle certification l
abel attached on the driver's (or front passenger's side center pillar gives the vehic
le identification number (VTN).nnTIRE SPECIFICATION AND PRESSURE LABELnnThe tires supp
<

chain.invoke("What are the SAFETY AND VEHICLE DAMAGE WARNING")

'This manual includes information titled as DANGER, WARNING, CAUTION and NOTICE. These
titles indicate the following:nnDANGERnnDANGER indicates a hazardous situation which,
if not avoided, will result in death or serious injury.nnAwnnnWARNINGnnWARNING indicate
s a hazardous situation which, if not avoided, could result in death or serious injur
y.nncautonnnCAUTIONnnCAUTION indicates a hazardous situation which, if not avoided, co
uld result in minor or moderate injury.nnNOTICE NOTICE indicates a situation which. if

chain.invoke("What are the switches are used to control")

chain.invoke("What are the features of the vehicle")

chain.invoke("What are the Specifications of Tires and Wheels")

```

```
chain.invoke("Which lubricants and fluids are recommended for use in your vehicle")

response=chain.invoke("What are the Specifications of Tires and Wheels")

from IPython.display import Markdown, display
display(Markdown(f"{response}"))

from IPython.display import Markdown, display, display_html, HTML, display_markdown
display(Markdown(response))

from IPython.display import Markdown, display
display(Markdown(f"{response[1]}"))
```

## ✓ Chatbot

```
#from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.text_splitter import CharacterTextSplitter, RecursiveCharacterTextSplitter
from langchain.vectorstores import DocArrayInMemorySearch
from langchain.document_loaders import TextLoader
from langchain.chains import RetrievalQA, ConversationalRetrievalChain
from langchain.memory import ConversationBufferMemory
from langchain.chat_models import ChatOpenAI
from langchain.document_loaders import TextLoader
from langchain.document_loaders import PyPDFLoader

retriever

def load_db(file,chain_type,k):
    loader = PyPDFLoader(file)
    documents = loader.load()
    # split documents
    text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=150)
    docs = text_splitter.split_documents(documents)
    # define embedding
    model_name="sentence-transformers/all-mpnet-base-v2"
    embeddings=HuggingFaceEmbeddings(model_name=model_name)
    # create vector database from data
    db = DocArrayInMemorySearch.from_documents(docs, embeddings)
    # define retriever
    retriever = db.as_retriever(search_type="similarity", search_kwargs={"k": k})
    LLM_Model="google/flan-t5-large"
    tokenizer=AutoTokenizer.from_pretrained(LLM_Model)
    model=AutoModelForSeq2SeqLM.from_pretrained(LLM_Model,torch_dtype=torch.float32)
    pipe=pipeline(
        "text2text-generation",
        model=model,
        tokenizer=tokenizer,
        max_length=300,
        temperature=0,
        top_p=0.95,
        repetition_penalty=1.15
    )
    llm=HuggingFacePipeline(pipeline=pipe)

    # create a chatbot chain. Memory is managed externally.
    qa = ConversationalRetrievalChain.from_llm(
        llm=llm,
        chain_type=chain_type,
        retriever=retriever,
        return_source_documents=True,
        return_generated_question=True,
    )
    return qa
```

```

import panel as pn
import param

class cbfs(param.Parameterized):
    chat_history = param.List([])
    answer = param.String("")
    db_query = param.String("")
    db_response = param.List([])

    def __init__(self, **params):
        super(cbfs, self).__init__(**params)
        self.panels = []
        self.loaded_file = "/content/manual.pdf"
        self.chain = load_db(self.loaded_file, "stuff", 4)

    def call_load_db(self, count):
        if count == 0 or file_input.value is None: # init or no file specified :
            return pn.pane.Markdown(f"Loaded File: {self.loaded_file}")
        else:
            file_input.save("temp.pdf") # local copy
            self.loaded_file = file_input.filename
            button_load.button_style="outline"
            self.qa = load_db("temp.pdf", "stuff", 4)
            button_load.button_style="solid"
            self.clr_history()
            return pn.pane.Markdown(f"Loaded File: {self.loaded_file}")

    def convchain(self, query):
        if not query:
            return pn.WidgetBox(pn.Row('User:', pn.pane.Markdown("", width=600)), scroll=True)
        result = self.chain({"question": query, "chat_history": self.chat_history})
        self.chat_history.extend([(query, result["answer"])])
        self.db_query = result["generated_question"]
        self.db_response = result["source_documents"]
        self.answer = result['answer']
        self.panels.extend([
            pn.Row('User:', pn.pane.Markdown(query, width=1000)),
            pn.Row('ChatBot:', pn.pane.Markdown(self.answer, width=1000, style={'background-color': '#F6F6F6'}))
        ])
        inp.value = '' #clears loading indicator when cleared
        return pn.WidgetBox(*self.panels, scroll=True)

    @param.depends('db_query', )
    def get_lquest(self):
        if not self.db_query :
            return pn.Column(
                pn.Row(pn.pane.Markdown(f"Last question to DB:", styles={'background-color': '#F6F6F6'})),
                pn.Row(pn.pane.Str("no DB accesses so far"))
            )
        return pn.Column(
            pn.Row(pn.pane.Markdown(f"DB query:", styles={'background-color': '#F6F6F6'})),
            pn.pane.Str(self.db_query )
        )

    @param.depends('db_response', )
    def get_sources(self):
        if not self.db_response:
            return
        rlist=[pn.Row(pn.pane.Markdown(f"Result of DB lookup:", styles={'background-color': '#F6F6F6'}))]
        for doc in self.db_response:
            rlist.append(pn.Row(pn.pane.Str(doc)))
        return pn.WidgetBox(*rlist, width=600, scroll=True)

    @param.depends('convchain', 'clr_history')
    def get_chats(self):
        if not self.chat_history:
            return pn.WidgetBox(pn.Row(pn.pane.Str("No History Yet")), width=600, scroll=True)
        rlist=[pn.Row(pn.pane.Markdown(f"Current Chat History variable", styles={'background-color': '#F6F6F6'}))]
        for exchange in self.chat_history:
            rlist.append(pn.Row(pn.pane.Str(exchange)))
        return pn.WidgetBox(*rlist, width=600, scroll=True)

    def clr_history(self, count=0):
        self.chat_history = []
        return

```



```
import os
import openai
import sys
sys.path.append('../..')

import panel as pn # GUI
pn.extension()
```



```
!pip install -U docarray
```



```
Requirement already satisfied: docarray in /usr/local/lib/python3.10/dist-packages (0.40.0)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from docarray) (1.25.2)
Requirement already satisfied: orjson>=3.8.2 in /usr/local/lib/python3.10/dist-packages (from docarray) (3.10.3)
Requirement already satisfied: pydantic>=1.10.8 in /usr/local/lib/python3.10/dist-packages (from docarray) (2.7.1)
Requirement already satisfied: rich>=13.1.0 in /usr/local/lib/python3.10/dist-packages (from docarray) (13.7.1)
Requirement already satisfied: types-requests>=2.28.11.6 in /usr/local/lib/python3.10/dist-packages (from docarray) (2.32.0.20240523)
Requirement already satisfied: typing-inspect>=0.8.0 in /usr/local/lib/python3.10/dist-packages (from docarray) (0.9.0)
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic>=1.10.8->docarray) (0.7)
Requirement already satisfied: pydantic-core==2.18.2 in /usr/local/lib/python3.10/dist-packages (from pydantic>=1.10.8->docarray) (2.18)
Requirement already satisfied: typing-extensions>=4.6.1 in /usr/local/lib/python3.10/dist-packages (from pydantic>=1.10.8->docarray) (4)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=13.1.0->docarray) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=13.1.0->docarray) (2.16.1)
Requirement already satisfied: urllib3>=2 in /usr/local/lib/python3.10/dist-packages (from types-requests>=2.28.11.6->docarray) (2.0.7)
Requirement already satisfied: mpyy-extensions>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from typing-inspect>=0.8.0->docarray) (0.3.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich>=13.1.0->docarra
```

```
cb = cbfs()
```

```
file_input = pn.widgets.FileInput(accept='.pdf')
button_load = pn.widgets.Button(name="Load DB", button_type='primary')
button_clearhistory = pn.widgets.Button(name="Clear History", button_type='warning')
button_clearhistory.on_click(cb.clr_history)
inp = pn.widgets.TextInput(placeholder='Enter text here..')
```

```
bound_button_load = pn.bind(cb.call_load_db, button_load.param.clicks)
conversation = pn.bind(cb.convchain, inp)
```

```
jpg_pane = pn.pane.Image( './img/convchain.jpg')
```

```
tab1 = pn.Column(
    pn.Row(inp),
    pn.layout.Divider(),
    pn.panel(conversation, loading_indicator=True, height=700),
    pn.layout.Divider(),
)
tab2= pn.Column(
    pn.panel(cb.get_lquest),
    pn.layout.Divider(),
    pn.panel(cb.get_sources ),
)
tab3= pn.Column(
    pn.panel(cb.get_chats),
    pn.layout.Divider(),
)
tab4=pn.Column(
    pn.Row( file_input, button_load, bound_button_load),
    pn.Row( button_clearhistory, pn.pane.Markdown("Clears chat history. Can use to start a new topic" )),
    pn.layout.Divider(),
    pn.Row(jpg_pane.clone(width=400))
)
dashboard = pn.Column(
    pn.Row(pn.pane.Markdown('# ChatWithME')),
    pn.Tabs(('Conversation', tab1), ('Database', tab2), ('Chat History', tab3),('Configure', tab4))
)
dashboard
```



# ChatWithME

Conversation Database Chat History Configure

Enter text here...

User: What is Vehicle Identification Number(VIN)?

ChatBot: the number used in registering your car and in all legal matters pertaining to its ownership, etc

User: Where could i find it?

ChatBot: The number is punched on the floor under the front right seat

User: Which lubricants and fluids are recommended for use in your vehicle?

ChatBot: Engine oil \*1 \*2 \*4 (drain and refill)Petrol 1.0 T-GDi ACEA C2 Petrol Smartstream G1.2 SAE 0W20, API SN PLUS/SP or ILSAC GF-6 Manual transmission fluid (Inte manual transmission fluid)Petrol 1.0 T-GDi aAPI GL-4, SAE 70W - SK : HK SYN MTF 70W- H.K.SHELL : SPIRAX S6 GHME 70W MTF- GS CALTEX : GS MTF HD 70WPetrol Smartstream G1.2 a Intelligent manual transmission system actuator fluidPetrol 1.0 T-GDi aAPI GL-4, SAE 70W 0W-30, 5W-30, 5W-401.0 T-GDiAdvanci G1.2

User: List some cautions of the vehicle.

ChatBot: Severe engine and transmission damage may result from the use of poor quality

User: what is the warning on fuels?

ChatBot: Damage to the fuel system or performance problem caused by the use of these fuels may not be covered by your New Vehicle Limited Warranty

User: what are those fuels

ChatBot: Petrol containing alcohol and methanol

```
/usr/local/lib/python3.10/dist-packages/langchain_core/_api/deprecation.py:119: LangChainDeprecationWarning: The method `Chain.__call__`  
warn_deprecated(  
Token indices sequence length is longer than the specified maximum sequence length for this model (578 > 512). Running this sequence th  
<ipython-input-24-552a5b105dc4>:38: PanelDeprecationWarning: 'style' is deprecated and will be removed in version 1.4, use 'styles' ins  
pn.Row('ChatBot:', pn.pane.Markdown(self.answer, width=1000, style={'background-color': '#F6F6F6'}))  
<ipython-input-24-552a5b105dc4>:38: PanelDeprecationWarning: 'style' is deprecated and will be removed in version 1.4, use 'styles' ins  
pn.Row('ChatBot:', pn.pane.Markdown(self.answer, width=1000, style={'background-color': '#F6F6F6'}))  
<ipython-input-24-552a5b105dc4>:38: PanelDeprecationWarning: 'style' is deprecated and will be removed in version 1.4, use 'styles' ins  
pn.Row('ChatBot:', pn.pane.Markdown(self.answer, width=1000, style={'background-color': '#F6F6F6'}))  
<ipython-input-24-552a5b105dc4>:38: PanelDeprecationWarning: 'style' is deprecated and will be removed in version 1.4, use 'styles' ins  
pn.Row('ChatBot:', pn.pane.Markdown(self.answer, width=1000, style={'background-color': '#F6F6F6'}))  
<ipython-input-24-552a5b105dc4>:38: PanelDeprecationWarning: 'style' is deprecated and will be removed in version 1.4, use 'styles' ins  
pn.Row('ChatBot:', pn.pane.Markdown(self.answer, width=1000, style={'background-color': '#F6F6F6'}))  
<ipython-input-24-552a5b105dc4>:38: PanelDeprecationWarning: 'style' is deprecated and will be removed in version 1.4, use 'styles' ins  
pn.Row('ChatBot:', pn.pane.Markdown(self.answer, width=1000, style={'background-color': '#F6F6F6'}))
```

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.