HSN Code Classifier

Description

The "HSN Code Classifier" is a specialized application designed to facilitate container clearance for Indian Customs. It accurately classifies products and assign the right product description based on the name written in a container. This application not only assigns the correct HSN code to each product but also provides justifications for why a specific product has been classified in a particular HSN code category. It is particularly valuable for customs clearance processes, where incoming containers often lack standardized HSN codes, making accurate classification essential for efficient cargo clearance.

Inside the application

Install libraries

Following libraries are pre-installed with this application

```
pip3 install astor more-itertools pathlib regex torch numpy pandas IPython langchain transformers torchvision llama-cpp-python faiss-gpu faiss-cpu --use-pep517
```

Import Libraries

Imports various libraries and modules into a Jupyter Notebook or Python script. Overall, this code block appears to prepare the environment for a diverse range of

data processing and natural language understanding tasks, including machine learning and text analysis.

In [ ]:
```
import ast
import atexit
import io
import itertools
import json
import os
import pathlib
import re
import sys
import time
import uuid
from datetime import datetime
import torch
import numpy as np
import pandas as pd
from IPython.display import HTML, display
from langchain import PromptTemplate
from langchain.chains import RetrievalQA
from langchain.document_loaders import DirectoryLoader, TextLoader
from langchain.document_loaders.csv_loader import CSVLoader
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.llms import CTransformers
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.vectorstores import FAISS
from llama_cpp import Llama
```

Select Device

Select the right device depending on it cuda or mps is available or not

In [ ]:
```
# Define the device (either "cuda" for GPU or "cpu" for CPU)
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
```

Read & Load the HSN Code

Provide the Path of the HSN Code master file

Note: *In later stage, there will be an upload capabilty to upload the file or even preload the master file during the notebook loading*

In [ ]:
```
from sentence_transformers import InputExample, SentenceTransformer, loss
hts_df = pd.read_csv("hts_flattened_data.csv")


texts = CSVLoader(
    file_path="./hts_flattened_data.csv", source_column="HTS Number"
).load()
model_name = "all-mpnet-base-v2"
model_kwargs = {"device": device}
encode_kwargs = {}
embeddings = HuggingFaceEmbeddings(
    model_name=model_name, model_kwargs=model_kwargs, encode_kwargs=encod
)
```

Prompt / Instruction to the AI/ML Model

Load the LLM Model and provide prompt/instruction to the model to anwer to your queries.

In [ ]:
```
from langchain.prompts import PromptTemplate
from langchain.llms import LlamaCpp
from langchain.chains import LLMChain
from langchain.callbacks.streaming_stdout import StreamingStdOutCallbackH
from langchain.callbacks.manager import CallbackManager
from langchain.llms import GPT4All, LlamaCpp

# create and save the local database
if not os.path.exists("faiss"):
    db = FAISS.from_documents(texts, embeddings)
    db.save_local("faiss")

model_name = "all-mpnet-base-v2"
model_kwargs = {"device": device}
encode_kwargs = {}
model_type = "llama"
# model_path = f"{os.getcwd()}/llama-2-7b.Q5_0.gguf"
model_path = f"/Users/bhowmick.sumit/AI_ML_ETL_BI_Models/Llama-2-7B-GGUF/
model_n_batch = 8
callbacks = []
# load embeddings
embeddings = HuggingFaceEmbeddings(
    model_name=model_name, model_kwargs=model_kwargs, encode_kwargs=encod
)
db = FAISS.load_local("faiss", embeddings)

# kNN on LLM content
retriever = db.as_retriever(search_kwargs={"k": 2})
llm = LlamaCpp(model_path=model_path, n_ctx=2048,
               n_batch=model_n_batch, verbose=False, temperature=0.001)


# prepare the template we will use when prompting the AI
template = """As an expert cargo inspector, you will receive several cont

Product description will be provided as a query to you. Your responsibili

When you encounter a word in the description that is misspelled, try corr

If user terms or product descriptions don't make coherent sense, provide

In all the case, provide at least 3 suggestions in descending order of re

While showing the result, print each of the HTS Codes in a single line. W

HTS Code:
Description:
Exaplantion - why it is suited:

Context: {context}

Question: {question}
```

```
Helpful answer:
"""
prompt = PromptTemplate(template=template, input_variables=[
                        "context", "question"])

hts_llm = RetrievalQA.from_chain_type(
    llm=llm,
    chain_type="stuff",
    retriever=retriever,
    return_source_documents=True,
    chain_type_kwargs={"prompt": prompt},
)
```

Query and Response Layout

Instruct the model to read the query and answer as per the given prompt above

In [ ]:
```
def query(model, question):
    time_start = time.time()
    output = model({"query": question})
    response = output["result"]
    time_elapsed = time.time() - time_start
    display(
        HTML(f"<code>{model_name} response time: {time_elapsed:.02f} sec<
    print(response)
    return response
```

Know the HSN Code for a given Product Description

Please provide the product description to know the HSN Code

In [ ]:
```
query(
    hts_llm,
    "ww2, RAF, flying, helmet",
)
```

Another example: Please provide the product description to know the HSN Code

In [ ]:
```
query(
    hts_llm,
    "lip gloss t shirt",
)
```