**DS Lab**

**Exp-2**

**Name: Shashwat Tripathi**

**Division: D20A**

**Roll no: 64**

**Aim:** To build a Cognitive text based application to understand context for a Customer

service application/ Insurance/ Healthcare Application/ Smarter Cities/ Government

etc.

**Theory:**

**Cognitive Computing**

Cognitive computing refers to the simulation of human thought processes in a computerized model. It involves self-learning systems that use data mining, pattern recognition, and natural language processing to mimic the way the human brain works. Cognitive systems can analyze large amounts of unstructured data (like text, images, or videos) to extract insights and make decisions or predictions.

**Context Understanding**

Context understanding is a critical component of cognitive applications. It allows the system to grasp not just the literal meaning of words, but also the context in which they are used. This includes understanding:

User Intent: Determining what the user wants to achieve.

Entity Recognition: Identifying key entities like names, dates, locations, or specific terms related to a domain.

Sentiment Analysis: Gauging the emotional tone of the conversation.

Conversation Flow: Keeping track of the dialogue's flow to maintain coherence and relevance across multiple exchanges.

**Key Components for Building Cognitive Applications**

**Natural Language Processing (NLP):**

Tokenization: Breaking down text into individual words or phrases.

Part-of-Speech Tagging: Identifying the grammatical parts of speech in a sentence.

Named Entity Recognition (NER): Extracting entities like names, places, dates, etc.

Sentiment Analysis: Determining the sentiment behind the text (positive, negative, neutral).

Text Classification: Categorizing text into predefined categories (e.g., spam vs. non-spam).

**Machine Learning:**

Supervised Learning: Training models with labeled data to predict outcomes.

Unsupervised Learning: Finding patterns in data without predefined labels.

Reinforcement Learning: Models learn to make decisions by receiving feedback (rewards or penalties).

**Domain-Specific Knowledge:**

Customer Service: Understanding common customer inquiries and troubleshooting processes.

Insurance: Recognizing insurance terminology and processing claims.

Healthcare: Interpreting medical terminology and patient records.

Smarter Cities: Integrating data from various urban systems for efficient city management.

Government: Processing and responding to public inquiries, legal documents, and policy information.

**Code:**

from transformers import pipeline

import spacy

from google.colab import files

# Load spaCy model

nlp = spacy.load('en\_core\_web\_sm')

# Initialize pipelines

summarizer = pipeline('summarization')

qa\_pipeline = pipeline('question-answering')

# Upload text file

uploaded = files.upload()

# Read and analyze the lesson plan

file\_name = list(uploaded.keys())[0]

with open(file\_name, 'r') as file:

lesson\_text = file.read()

# Process the text with spaCy

doc = nlp(lesson\_text)

key\_concepts = [chunk.text for chunk in doc.noun\_chunks if chunk.root.dep\_ == "nsubj"]

# Summarize the lesson

summary = summarizer(lesson\_text, max\_length=50, min\_length=25, do\_sample=False)

summary\_text = summary[0]['summary\_text']

print("Key Concepts:", key\_concepts)

print("Summary:", summary\_text)

# Function to answer questions based on the text

def answer\_question(question):

result = qa\_pipeline(question=question, context=lesson\_text)

return result['answer']

# Example usage

while True:

user\_question = input("Ask a question related to the text (or type 'exit' to quit): ")

if user\_question.lower() == 'exit':

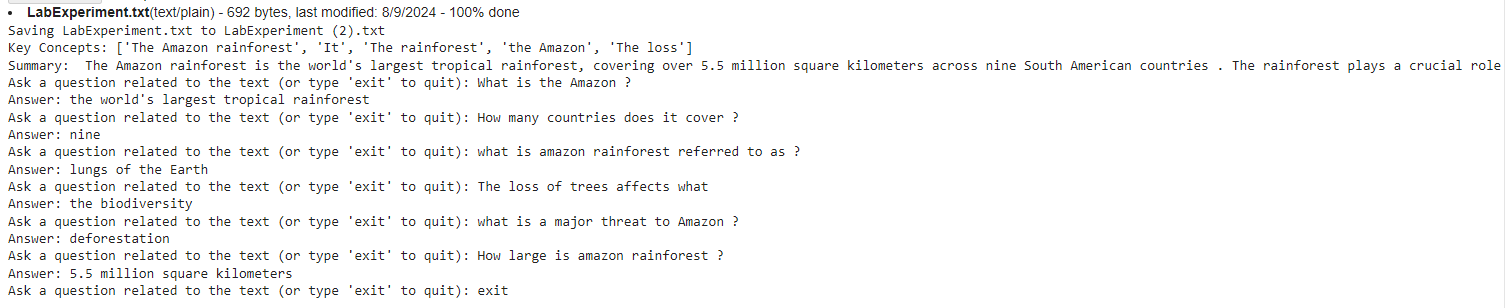
break

answer = answer\_question(user\_question)

print("Answer:", answer)

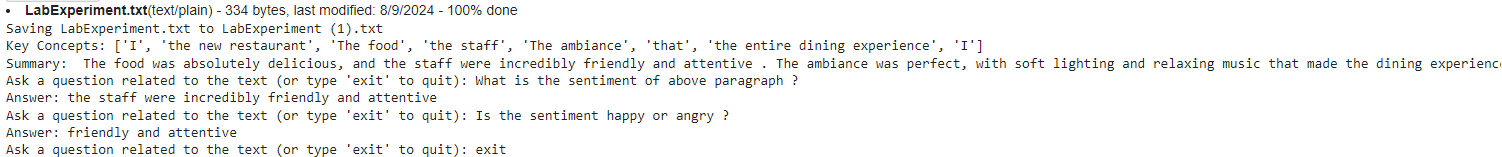
**Output 1 :**

Input 1 was a paragraph about Amazon rainforests. The model understood the context and answered all the questions correctly.



**Output 2 :**

Input 2 was a paragraph to check the sentiment analysis of model. The output shows that the model works perfectly fine.



**Conclusion:** Therefore cognitive text-based applications represent the next step in making interactions between humans and machines more natural and effective.