

**Aim:** To build a Cognitive text based application to understand context for a Customer service application/ Insurance / Healthcare Application / Smarter Cities / Government etc.

### Theory:

#### **What is a Cognitive Text-Based Application?**

A cognitive text-based application is an intelligent software system that can understand and respond to human language in a meaningful way. These applications use Natural Language Processing (NLP), Machine Learning (ML), and AI techniques to process user input and provide accurate, human-like responses.

#### **What is Cognitive Computing?**

Cognitive computing is a branch of Artificial Intelligence (AI) that tries to mimic how humans think, learn, and make decisions. It involves:

- Understanding language (using NLP)
- Learning from experience (Machine Learning)
- Making decisions or predictions based on data

These systems are able to analyze large amounts of data, recognize patterns, and provide helpful responses — just like a human would in real-world scenarios.

#### **What is Natural Language Processing (NLP)?**

Natural Language Processing (NLP) is a field of AI that allows computers to understand, interpret, and respond to human language. It includes:

- Tokenization – Breaking text into words or sentences
- Stopword Removal – Ignoring common words like “the”, “is”, etc.
- Lemmatization/Stemming – Converting words to their root forms
- Entity Recognition – Identifying names, places, numbers, etc.

In this experiment, we use spaCy and nltk libraries for NLP tasks.

#### **Use of Cognitive Applications in Different Sectors**

Sector	Use Case
Healthcare	Disease diagnosis, patient monitoring, personalized medicine
Insurance	Claim support, fraud detection, premium assistance
Finance	Risk management, customer support, fraud detection
Retail	Personalized shopping, customer assistance, supply chain optimization
Education	Personalized learning, student performance tracking
Government	Public service queries, policy analysis, emergency/crisis management
Smarter Cities	Electricity usage, traffic control, pollution tracking, citizen help

#### **What is Keyword-Based Analysis?**

In simple applications, we use keyword matching where the system checks if certain important words (keywords) are present in the user's input. Based on that, it gives a predefined response.

Example:

- If a user says: "How to save energy?", the system detects the keyword "save" and "energy" and replies with saving tips.

This is a simple yet effective way to build chatbot-style applications.

#### Technologies/Libraries Used:

Library	Purpose
<b>spaCy</b>	Advanced NLP: tokenization, stopwords removal, lemmatization
<b>nlTK</b>	Basic NLP: tokenization, stopwords list, text processing utilities
<b>Python</b>	Programming language for logic and interaction

#### Steps in Building the Cognitive Text-Based Application:

1. Data Preparation  
Prepare a list of common user queries and their expected responses (domain-specific).
2. Text Preprocessing  
Clean user input using NLP: lowercase conversion, stopwords removal, tokenization.
3. Keyword Matching  
Check the user's input for keywords to determine what type of question is being asked.
4. Classification and Response Generation  
Return a matching response or a default message if the input is not recognized.
5. User Interaction  
The chatbot interacts with the user continuously until the user exits (e.g., says "bye").

```
# Install required libraries
!pip install spacy --quiet
!python -m spacy download en_core_web_sm

# Download NLTK resources
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('punkt_tab') # Added download for punkt_tab

# Load libraries
import spacy
nlp = spacy.load("en_core_web_sm")

# Sample queries and responses related to smart electricity usage
queries_and_responses = [
    ("electricity bill", "Your electricity bill this month is ₹1350. Would you like tips to reduce it?"),
    ("high power usage", "High usage detected in air conditioning during peak hours."),
    ("suggestions to save", "Try turning off appliances completely instead of standby mode."),
    ("energy saving", "Switch to LED bulbs and energy-efficient fans for better savings."),
    ("last month usage", "You used 210 units last month, mostly from kitchen and HVAC."),
    ("standby power", "Devices in standby mode still consume electricity. Unplug them."),
    ("check appliance usage", "Washing machine used 30 units this month."),
    ("solar energy", "You can save 40% on bills by installing rooftop solar panels."),
    ("daily consumption", "Today's consumption so far: 4.5 units."),
]

# Default response types
default_responses = {
    "greeting": "Hello! I'm your Smart Home Electricity Assistant. Ask me anything!",
    "farewell": "Goodbye! Stay energy-efficient!",
    "default": "Sorry, I didn't understand that. Try asking about electricity bills, savings, or appliance usage."
}
```

```
# Function to classify and respond
def classify_query(user_query):
    user_query = user_query.lower()
    user_tokens = set(nltk.word_tokenize(user_query))

    if user_tokens.intersection({"hi", "hello", "hey"}):
        return "greeting"
    elif user_tokens.intersection({"bye", "goodbye"}):
        return "farewell"

    for keywords, response in queries_and_responses:
        keyword_tokens = set(keywords.split())
        if user_tokens.intersection(keyword_tokens):
            return response

    return "default"

# Main chatbot loop
def chatbot():
    while True:
        user_query = input("You: ")
        response_type = classify_query(user_query)

        if response_type == "greeting":
            print("Assistant:", default_responses["greeting"])
        elif response_type == "farewell":
            print("Assistant:", default_responses["farewell"]) # Typo
        elif response_type != "default":
            print("Assistant:", response_type)
        else:
            print("Assistant:", default_responses["default"])

    break

# Run the chatbot
if __name__ == "__main__":
    chatbot()
```

**Output:**

```
You: Hello
Assistant: Hello! I'm your Smart Home Electricity Assistant. Ask me anything!
You: Why is my electricity bill high?
Assistant: Your electricity bill this month is ₹1350. Would you like tips to reduce it?
You: Suggest energy saving tips
Assistant: Switch to LED bulbs and energy-efficient fans for better savings.
You: Bye
Assistant: Goodbye! Stay energy-efficient!
```

**Conclusion:**

Cognitive text-based applications play a vital role in automating customer service, providing personalized assistance, and improving efficiency in various domains. Using NLP techniques and keyword analysis, even a simple chatbot can become a powerful tool to interact with users effectively. This experiment demonstrates the foundation of such intelligent systems using Python, spaCy, and nltk.

[AI&DS2\\_Expt\\_02](#)