Name-Shrusti Chintawar
Roll no -09
Batch C

EXPERIMENT NO – 2

Aim: To build a Cognitive text based application to understand context for Insurance Help.

Theory:

Objective:
The objective of this experiment is to develop a basic cognitive text-based application that can assist users with insurance-related queries. The application will be able to understand user input, identify keywords, and provide relevant information or responses. The primary goal is to demonstrate the basic principles of building a text-based chatbot for insurance help.

Theoretical Background:
1. **Natural Language Processing (NLP):** NLP is a field of artificial intelligence that focuses on the interaction between computers and human language. It includes techniques for text analysis, sentiment analysis, and language understanding, which are essential for building chatbots.

2. **Keyword-Based Text Analysis:** Keyword-based text analysis involves searching for specific words or phrases in a text to extract relevant information. In this experiment, we will use keyword matching to identify user queries related to insurance.

Libraries Used:
- **spaCy:** We will use the spaCy library for advanced text processing, including tokenization, lemmatization, and stop word removal.
- **nltk** - NLTK (Natural Language Toolkit) is a popular Python library for natural language processing, offering a wide range of tools and resources for tasks such as text analysis, tokenization, and part-of-speech tagging.

Steps:

1. Data Preparation: We will define a set of sample customer queries and their corresponding insurance responses. These queries will be used to train and test our chatbot.

2. Text Preprocessing: We will preprocess the text by tokenizing, lemmatizing, and removing stop words and punctuation using spaCy.

3. Keyword Matching: We will implement keyword matching to identify relevant queries based on the presence of specific keywords.

4. User Interaction: Users will interact with the chatbot by entering queries, and the chatbot will respond based on the identified keywords and similarity scores.

Expected Outcome:

The experiment is expected to result in a basic insurance assistance chatbot that can respond to user queries by matching them with predefined keywords and responses. It will demonstrate the fundamental principles of building a cognitive text-based application for insurance help.

Code:

```
import nltk
import spacy

# Load spaCy model
nlp = spacy.load("en_core_web_sm")

# Sample student queries and responses with keywords
queries_and_responses = [
    ("exam schedule", "The exam schedule is available on the university portal."),
    ("semester results", "Semester results will be declared next week."),
    ("assignment deadline", "The assignment deadline is 10th August."),
    ("marksheet request", "You can request your marksheet via the academic section."),
    ("revaluation process", "The revaluation process starts 2 days after results."),
    ("attendance percentage", "You can check your attendance in the student dashboard."),
```

```
    ("course registration", "Course registration starts from 15th
August."),
    ("internship certificate", "Submit your internship report to get the
certificate."),
    ("apply for leave", "Apply for leave through the college management
system."),
]

# Default responses
default_responses = {
    "greeting": "Hi there! How can I help you regarding student services?",
    "farewell": "Bye! Good luck with your studies.",
    "default": "Sorry, I didn't understand that. Could you ask
differently?",
}

# Function to classify user queries
def classify_query(user_query):
    user_query = user_query.lower()

    # Greetings & farewells
    if any(greeting in user_query for greeting in ["hi", "hello", "hey"]):
        return "greeting"
    elif any(farewell in user_query for farewell in ["bye", "goodbye", "see
you"]):
        return "farewell"

    # Keyword matching
    for keywords, response in queries_and_responses:
        for keyword in keywords.split():
            if keyword in user_query:
                return response

    return "default"

# Interactive loop
while True:
    user_query = input("You: ")

    query_type = classify_query(user_query)

    if query_type == "greeting":
        print("Chatbot:", default_responses["greeting"])
    elif query_type == "farewell":
```

```
        print("Chatbot:", default_responses["farewell"])
        break
    elif query_type != "default":
        print("Chatbot:", query_type)
    else:
        print("Chatbot:", default_responses["default"])
```

-Output:

```
You: hi
Chatbot: Hi there! How can I help you regarding student services?
You: When is the exam schedule released?
Chatbot: The exam schedule is available on the university portal.
You: bye
Chatbot: Bye! Good luck with your studies.
```

Conclusion:

In this experiment, we successfully developed a basic cognitive text-based application for insurance help, showcasing the principles of keyword-based text analysis and response generation.