

AIDS 2 Lab

Name: Rushabh Jain Class: D20B Roll No: 23

✓ Aim

To build a cognitive text-based application to assist users with healthcare-related queries using NLP techniques.

THEORY

This experiment focuses on constructing a basic cognitive chatbot for healthcare inquiry resolution. The chatbot understands user input, extracts medical-related keywords, and provides appropriate responses. This demonstrates foundational NLP principles applied to the healthcare domain.

Theoretical Background

- **Natural Language Processing (NLP):**
Facilitates communication between machines and human language, covering syntax, semantics, and text classification.
 - **Keyword-Based Text Analysis:**
Extracts significant terms from text. Utilized here for identifying symptoms, conditions, or treatment-related terms in queries.
-

Libraries Used

- **spaCy**
Tokenization, lemmatization, entity recognition, and stop word removal.
 - **nltk**
Intent classification and auxiliary token processing.
-

Steps

1. **Data Preparation:**
Create query-response pairs related to basic healthcare topics (e.g., fever, headache, appointment scheduling).

2. Text Preprocessing:

Clean user input using spaCy's tokenizer and lemmatizer.

3. Keyword Matching:

Detect medical keywords using direct match or similarity threshold.

4. User Interaction:

Command-line interface to take user queries and return appropriate responses.

Expected Outcome

A working chatbot capable of handling elementary healthcare queries, demonstrating practical use of NLP in healthcare conversational systems.

```
import spacy
import nltk
from nltk.tokenize import word_tokenize

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

# Sample data
sample_queries = {
    "fever": "For fever, it is advised to stay hydrated and take paracetamol.",
    "headache": "For headache, rest in a quiet, dark room and avoid screens.",
    "appointment": "You can book an appointment by calling our front desk at 123-456.",
    "cold": "For cold symptoms, take rest and drink warm fluids.",
    "cough": "Cough can be treated with cough syrup and warm tea.",
    "thank" : "My Pleasure.",
    "care" : "Connecting you with our customer support."
}

def preprocess_text(text):
    doc = nlp(text.lower())
    return [token.lemma_ for token in doc if not token.is_stop and not token.is_punct]

# Intent classification
def classify_intent(user_input):
    tokens = preprocess_text(user_input)
    for word in tokens:
        if word in sample_queries:
            return word
    return None

def generate_response(user_input):
    intent = classify_intent(user_input)
    if intent:
        return sample_queries[intent]
    return "Sorry, I couldn't understand your query."
```

```
if __name__ == "__main__":  
    while True:  
        user_query = input("Enter your health query (type 'exit' to quit): ")  
        if user_query.lower() == 'exit':  
            break  
        print("Bot:", generate_response(user_query))
```

↩ Enter your health query (type 'exit' to quit): I have fever
Bot: For fever, it is advised to stay hydrated and take paracetamol.
Enter your health query (type 'exit' to quit): I want to book an appointment
Bot: You can book an appointment by calling our front desk at 123-456.
Enter your health query (type 'exit' to quit): okay thank you
Bot: My Pleasure.
Enter your health query (type 'exit' to quit): exit

✓ Conclusion

This experiment demonstrates the construction of a basic cognitive chatbot for healthcare-related queries using spaCy and nltk. The chatbot employs preprocessing techniques such as tokenization, lemmatization, and stop word removal to clean user input. Intent classification through keyword matching enables the system to associate queries with predefined responses. The implementation validates foundati

Double-click (or enter) to edit