NLP Chatbot README

1 Introduction

This project implements a simple Natural Language Processing (NLP) chatbot using Python and the NLTK library. The chatbot processes user input, responds to basic greetings, and provides relevant responses based on a predefined corpus using TF-IDF vectorization and cosine similarity.

2 Features

- Responds to greetings like "hello", "hi", "hey", etc.
- Processes user queries using NLP techniques (tokenization, lemmatization, TF-IDF).
- Matches user input to the most relevant response from a predefined corpus.
- Exits the chat loop when the user types "bye".

3 Requirements

- Python 3.x
- NLTK library (`pip install nltk`)
- scikit-learn library (`pip install scikit-learn`)

4 Installation

1. Clone or download the repository.
2. Install the required dependencies:

   ```
   pip install nltk scikit-learn
   ```

3. Run the script to download necessary NLTK resources (punkt and wordnet) automatically.

5 Usage

1. Run the script (`NLP_Chatbot.py`):

   ```
   python NLP_Chatbot.py
   ```

2. Interact with the chatbot:
   - Type a greeting (e.g., "hello") to receive a friendly response.
   - Ask questions or provide input, and the chatbot will respond based on the corpus.
   - Type "bye" to exit the chat.

6 How It Works

- **Preprocessing**: The input text is tokenized, lemmatized, and normalized using NLTK's WordNet lemmatizer and punctuation removal.
- **Greeting Detection**: Checks for greeting keywords and responds with a random greeting from a predefined list.
- **Response Generation**: Uses TF-IDF vectorization and cosine similarity to find the most relevant response from the corpus.
- **Corpus**: A small, predefined text corpus is used for responses. You can expand it by modifying the `corpus` variable in the script.

7 Example Interaction

```
Chatbot: My name is NLPBot. Type 'bye' to exit.
You: hello
Chatbot: Hey there!
You: what can you do?
Chatbot: I can answer your basic questions using NLP.
You: bye
Chatbot: Bye! Take care.
```

8 Limitations

- The chatbot relies on a small, static corpus, limiting its knowledge base.
- Responses may not always be accurate for complex or out-of-context queries.
- No support for advanced conversational flows or external data sources.

9 Future Improvements

- Expand the corpus with more diverse responses or load it from an external file.
- Integrate a more sophisticated NLP model for better understanding.
- Add support for context-aware conversations or external APIs for dynamic responses.